A Bayesian Approach to Estimating Mallard Duckling Survival:
Understanding the Relationships with Wetland Density and Hatch Date

Wilson Wright

Department of Mathematical Sciences
Montana State University

April 27, 2016

A writing project submitted in partial fulfillment
of the requirements for the degree

Master of Science in Statistics

APPROVAL

of a writing project submitted by

Wilson Wright

This writing project has been read by the writing project advisor and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the Statistics Faculty.

Date	Megan Higgs Writing Project Advisor
Date	Mark Greenwood Writing Projects Coordinator

Abstract

Mallard (*Anas platyrhynchos*) duckling survival is an important, but not well understood, component of reproductive success for the species. To further investigate how mallard duckling survival is related to hatch date and wetland density, we analyze duckling counts for broods in southwestern Manitoba during the late 1980's. Various models are explored to explain the patterns in the observed data and these approaches are compared to the original analysis (Rotella and Ratti 1992). We demonstrate fitting models using a Bayesian approach and discuss the advantages of this framework in comparison to maximum likelihood estimation. Different Bayesian models are compared to one another and evaluated using posterior predictive checks. Zero-inflated binomial models appear to provide an ecologically realistic description of these data and allow a more complete understanding of how duckling survival is related to wetland density and hatch date.

Contents

1 Introduction **3**

2 Methods **4**
2.1 Data Collection 4
2.2 Data Cleaning 5
2.3 Analysis 6

3 Exploratory Plots **10**

4 Analysis **17**
4.1 Maximum Likelihood Approach 17
4.2 Bayesian Logistic Regression 18
4.2.1 Model 18
4.2.2 Results 19
4.2.3 Posterior Predictive Checks 21
4.3 Accounting for Overdispersion using Normal Errors 26
4.3.1 Model 26
4.3.2 Results 29
4.3.3 Posterior Predictive Checks 31
4.4 Beta-Binomial Model for Overdispersion 34
4.5 Zero-Inflated Binomial Approach 36
4.5.1 Model 36
4.5.2 Results 38
4.5.3 Posterior Predictive Checks 40
4.5.4 Residual Plots 44
4.6 Zero-Inflated Binomial Model with Covariates for π 50
4.6.1 Model 50
4.6.2 Results 51
4.6.3 Residual Plots 54

5 Discussion **56**
5.1 Additional Inferences 56
5.2 Comparison to other Duckling Survival Analyses 60
5.3 Conclusion 62

References **65**

A Appendix **66**
A.1 Plots 66
A.2 R Code 70
A.3 Stan Model Code 89

1 Introduction

Understanding the factors influencing the reproductive success of a species is of interest to many ecologists and can have important implications for wildlife management and/or conservation efforts. To investigate factors associated with reproductive success in mallard ducks (*Anas platyrhynchos*), we reanalyze data collected and first analyzed by Rotella and Ratti (1992) during the late 1980's. The original study examined how mallard recruitment (i.e., survival of juveniles) was associated with habitat conditions and brood hatch date for the three years data were collected. At the time of the original study, there were concerns that mallard nesting success had declined due to habitat loss but there was little research investigating this relationship (Rotella and Ratti 1992). This paper will further examine mallard reproductive success and explore how more sophisticated models can help researchers better understand duckling survival.

The original analysis of these data focused on examining brood success (having at least one surviving duckling) using binary logistic regression. That analysis indicated that "brood survival was directly related to wetland density and inversely related to hatching date" (Rotella and Ratti 1992). Additionally, median tests were used to compare duckling survival (proportion surviving after thirty days) among categories of wetland density and hatch date. While these covariates were treated is continuous in the binary logistic regression models, each of the variables were divided into a high and low level when examining duckling survival with the median tests. This paper reanalyzes these data with a focus on the probability of a duckling surviving and how it is related to these variables. We explore modeling the number of surviving ducklings in each brood as binomial counts where the probability of duckling survival for each brood is modeled using logistic regression with hatch date and number of ponds around the nest as covariates. Additional models are also examined as ways to account for observed patterns in the data and provide further insight about how

duckling survival probabilities are related to habitat conditions. We utilize a Bayesian approach to fit these models and evaluate them using posterior predictive checks. We also compare the inferences from these more sophisticated analyses to those from the original analysis.

2 Methods

2.1 Data Collection

Duckling survival information was collected for a total of 69 broods (each brood was associated with a single nest) in southwestern Manitoba during the years 1987 (10 broods), 1988 (21 broods), and 1989 (38 broods). Mallard hens were trapped on their nests using handled dip nets, hand-carried mist nets, purse traps or automatic nest traps. All hens were equipped with transmitters and subsequently monitored via radio telemetry. Once ducklings hatched, the number of ducklings and hatch date were recorded. Every brood was monitored for thirty days after its hatch date and the total number of surviving ducklings after this period was recorded for each brood. Therefore, these data provide information on how many ducklings were born and how many survived for at least thirty days from each brood, but does not include how or when ducklings died during the observed time period. Survival probability for ducklings is estimated to be high after thirty days of survival (Talent, Jarvis, and Krapu 1983; Rotella and Ratti 1992), meaning these data should provide useful information about mallard recruitment. The number of ducklings surviving after 30 days (y_i) out of the total ducklings born (m_i) are initially assumed to be binomially distributed for each brood (i) in this analysis.

The original study also collected habitat information for the entire study area each year and for an area around each hen's nest site. This paper focuses on using the habitat information specific to each brood in order to model the probability of

duckling survival for 30 days from each brood (p_i). The habitat data were collected for the area within a radius of 0.8 km around the nesting location within one week of that brood's hatch date. For each brood, the total number, total area, total perimeter, and type (i.e., permanent, semi-permanent, man-made) of all the ponds within the designated area were recorded. Further details on the data collection can be found in the original paper (Rotella and Ratti 1992). The original analysis compared different binary logistic regression models for brood survival using likelihood-ratio tests. The explanatory variables for the candidate models were selected based on which variables had strong evidence of non-zero coefficients in univariate analyses and also provided unique information. For example, Rotella and Ratti (1992) found that total number, total area, and total perimeter of ponds around each nest all provided nearly identical information about wetland density based on the high Spearman's correlations among the pairwise comparisons of these variables. This paper will focus on understanding how the total number of ponds and the hatch date for each brood are related to the probability of duckling survival. Exploratory plots illustrate that these covariates appear to explain heterogeneity in duckling survival probabilities among broods.

2.2 Data Cleaning

The original data file received for this analysis included a total of 98 broods, but examination of these data showed that some rows had been duplicated. The original paper describes 69 broods and when the duplicated rows were removed (after confirming that this was an error), the number of broods in the data file matched the description from the original analysis. It appears that since the original analysis, the data file had been altered to include duplicate rows for each brood that had at least one surviving duckling (Table 1). This error was originally noticed because some of the radio collar frequencies (freq) were listed twice within the same year of the study.

Table 1: Subset of rows from the original data file. These were rows for broods which had at least one surviving duckling and were all duplicated (one example is bolded). This pattern persisted for all broods with at least one surviving duckling.

freq	hatchd	year	eggs	count
786	173	1987	6	1
864	**162**	**1987**	**9**	**9**
1216	167	1987	10	3
1427	167	1987	9	1
786	173	1987	6	1
864	**162**	**1987**	**9**	**9**
1216	167	1987	10	3

2.3 Analysis

Generalized linear models (GLMs) can be used to analyze non-normally distributed response variables and investigate their relationships with covariates. Here, the number of surviving ducklings, y_i, out of the total hatched, m_i, in each brood are assumed to be binomially distributed random variables with the probability of survival for brood i equal to p_i, that is, $Y_i \sim \text{Binomial}(m_i, p_i)$. Logistic regression is a GLM frequently used for binomial responses and allows the inclusion of covariates to account for heterogeneity in the probability of duckling survival among broods using the logit link, where $\text{logit}(p_i) = \ln(p_i/(1-p_i))$. The logit link connects the systematic component of the model to the assumed binomially distributed response variable. The systematic component consists of a linear combination of explanatory variables (e.g., hatch date and number of ponds for these data) and can be described as

$$\text{logit}(p_i) = \beta_0 + \beta_1 x_{1i} + \ldots + \beta_q x_{qi},$$

where x_{1i} to x_{qi} are values of different variables for brood i and each β parameter is a regression coefficient to be estimated. This allows a different probability of survival for each brood depending on the values of the explanatory variables for that brood. Other link functions (e.g., probit) could be used in a GLM to analyze these data as

well, but the logit link is the canonical link for binomial responses (giving it desirable statistical properties) and provides easier interpretations of the regression coefficients in comparison to other links. Typically, all of the commonly used link functions will give similar fitted probabilities when analyzing binomial responses.

For logistic regression, estimates of regression coefficients (β parameters) are often found using maximum likelihood (ML) estimation. This approach identifies the combination of values of the parameters maximizing the likelihood function given the observed data. Assuming counts are binomially distributed, the likelihood function for this model is

$$L(\boldsymbol{\beta} \mid \mathbf{y}, \mathbf{m}) = \prod_{i=1}^{N} \binom{m_i}{y_i} p_i^{y_i} (1 - p_i)^{m_i - y_i}, \tag{2.1}$$

where each p_i is modeled using the logit link as described above. There are many statistical software packages available for estimating regression coefficients and associated standard errors using this approach. In this paper, we will use the `glm` function from Program R (R Core Team 2016) to demonstrate a ML approach for analyzing these data. When using ML estimation, standard deviations for parameters can be estimated using the likelihood function and used to construct Wald-type confidence intervals based on the asymptotic normality of ML estimates. Alternatively, profile likelihood confidence intervals can be constructed based directly on the curvature of the profile likelihood function for each parameter.

Instead of using ML estimation, a Bayesian approach can also be used to make inference about the regression coefficients. The likelihood function (Equation 2.1) describes the relative weight of evidence for the possible values of a parameter given the observed data, but is not a probability distribution for the parameters of interest. Comparatively, a Bayesian approach does obtain a posterior probability distribution for the parameters. In general notation, Bayes Theorem is expressed as

$$f(\theta \mid y) = \frac{f(y \mid \theta) f(\theta)}{f(y)}$$

and provides the theoretical foundation of Bayesian methods where interest is in combining a likelihood function $f(y \mid \theta)$ and specified prior distribution $f(\theta)$ to form a posterior distribution $f(\theta \mid y)$ for all model parameters. The posterior distribution of the parameters are described in various ways for inference. Posterior means are typically used as point estimates, but the mode or median of the posterior distribution can be used as well. When the prior for a parameter places uniform density over all possible values, the posterior mode will be equivalent to the ML estimate. Quantiles of the posterior distribution can be used to construct posterior intervals (PIs).

A Bayesian approach can provide numerous advantages to a data analysis. While prior specifications receive criticism from some (e.g., Lele and Dennis 2009), they provide an opportunity to incorporate any available researcher expertise into an analysis or a straightforward way to update the inferences of previous analyses. Additionally, the utilization of priors can improve estimation of parameters in some scenarios (e.g., complete separation, small sample sizes) where ML estimates will be imprecise and/or not useful. Interpretation of results from a Bayesian analysis are also frequently more intuitive than those from other approaches as well. For instance, a 90% PI is interpreted as having a 90% probability of including the parameter of interest. Comparatively, the concept of 'confidence' relies on the scenario (or thought experiment) of repeating the study many times and generating a confidence interval for each - where 90% of these 90% confidence intervals would include the true parameter. Researchers often want to interpret confidence intervals as the interval with the specified probability of including the parameter (as is the case for PIs), but this is incorrect despite being a natural way to think about them. Finally, while estimates of uncertainty from a ML approach to GLMs rely on asymptotic results, no such assumption is required for a Bayesian analysis. Interpretations of the

resulting PIs are valid regardless of sample size. Comparisons of inferences from ML and Bayesian approaches for these data will be shown throughout this report and illustrate some advantages of a Bayesian approach in this case.

Obtaining the posterior distribution for the parameters in a Bayesian model is often computationally intensive. For all but the most trivial examples, Markov chain Monte Carlo (MCMC) algorithms are used to approximate posterior distributions by obtaining samples from them. When using MCMC samples for inference, it is important to assess convergence of the algorithm in order to help ensure the collection of samples are a good approximation of the posterior distribution of interest. Convergence is typically assessed using the Gelman-Rubin statistic ($\hat{R}$) and/or traceplots, although many other diagnostics are available as well. Generally, $\hat{R}$ values less than 1.10 or 1.05 are considered an indication of convergence. When examining traceplots, convergence is suggested when every chain covers the same range of values and is rapidly moving through the parameter space. Traceplots with these characteristics are typically described as showing good 'mixing'. The Bayesian analyses in this report are performed using Stan (Carpenter et al. 2016) through R using the `rstan` package (Stan Development Team 2016).

Before fitting models, exploratory plots are examined to better understand the potential relationships between the available covariates and the probability of a duckling surviving for thirty days. For logistic regression, the logits of the observed proportion of ducklings surviving in each brood provide a way to assess the potential relationships in the survival probabilities with the covariates on this scale. In logistic regression, probabilities are modeled on the logit scale using a linear combination of covariates, meaning these plots can help inform what model structure could be useful in the analysis. However, since the logit transformation of proportions equal to zero and one are defined as $-\infty$ and ∞ respectively, to visualize these patterns we use empirical logits to create these plots. Empirical logits can be

calculated by adding a small amount to the number of 'successes' and 'failures': $\ln((y_i+0.5)/(m_i-y_i+0.5))$ (Ramsey and Schafer 2012). Data formatting is performed using the `dplyr` package (Wickham and Francois 2015) and we create plots using the `ggplot2` package (Wickham 2009). The R output in this report is displayed using the `xtable` package (Dahl 2015).

3 Exploratory Plots

In this analysis, the number of ponds around each nest site and hatch date are the two explanatory variables being used to model the duckling survival probabilities for each brood. Initially, these data are examined with plots in order to explore how the number of surviving ducklings in each brood is related to these explanatory variables and to help inform the structure of the logistic regression model. Note that the number of broods each year increased as the study progressed (Figure 1), potentially due to improvements in the techniques used to capture hens. Habitat conditions also differed considerably over the three years and this is discussed in the original paper. The wettest conditions were in 1988, while 1989 was noticeably drier

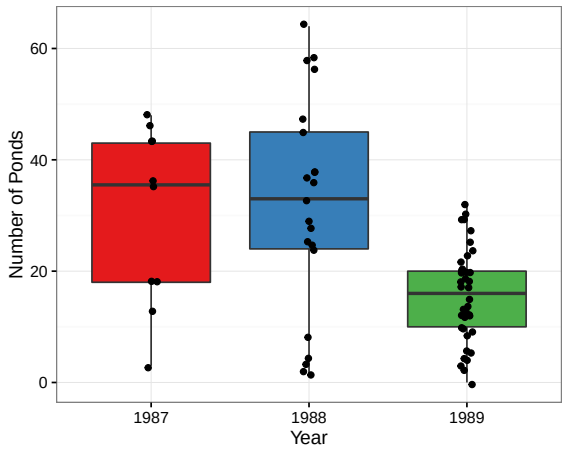

Figure 1: Boxplots of number of ponds surrounding each nest for each year illustrate that the first two years of the study (1987 and 1988) had similar, wetter conditions on average compared to 1989.

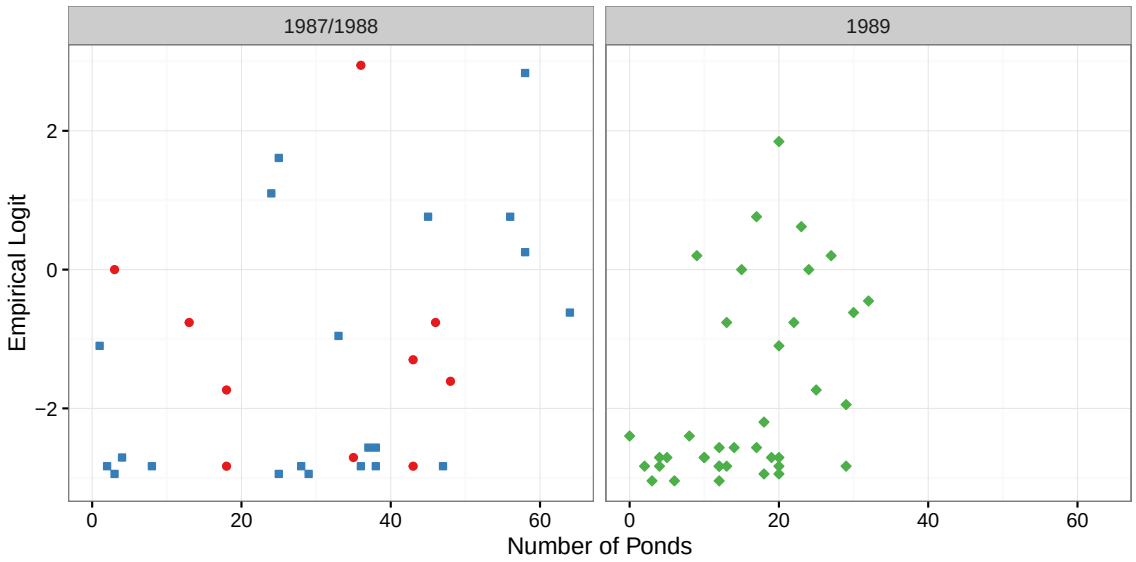

Figure 2: Empirical logits versus number of ponds for the first two years (1987 and 1988) compared to the final, drier year (1989). The relationship between duckling survival and number of ponds does not appear to change between years (although this is difficult to assess) even though there were no broods with more than 35 ponds in the vicinity in 1989. Points are coded by year - red circles for 1987, blue squares for 1988, and green diamonds for 1989.

in comparison to the first two years of the study (Figure 1), resulting in the number of ponds covariate being confounded with year. We will be making the assumption that the heterogeneity in duckling survival probability due to habitat conditions is accounted for with the number of ponds variable despite there also being some overall differences in habitat conditions each year. There is no way to assess whether this is reasonable since no broods in 1989 had large numbers of ponds which were observed in the first two years, but this assumption makes sense ecologically because nearby wetland density is expected to be a good surrogate for overall habitat quality for mallards.

To explore how the relationship between duckling survival probability and number of ponds differs by year, scatterplots of the empirical logits versus number of ponds for observations from 1987/1988 are compared to those for observations from 1989. We combine the observations from the first two years in this figure because

there are very few observations from 1987 and because the broods for the first two years have number of ponds values that are more similar to each other than to those from 1987 (Figure 1). While habitat conditions differed over these three years, it appears possible that the relationship between number of ponds and duckling survival probability is similar across years (Figure 2). This assessment is particularly difficult, if not impossible, to make for observations from 1989 because the broods had such a restricted range in the number of ponds. However, it seems reasonable to combine observations across years and assume this relationship does not change each year in order to make inference about an overall relationship between number of ponds and the probability of duckling survival. Overall, it appears that the probability of duckling survival increases as the number of ponds increases which is the anticipated relationship with this covariate.

Next, additional plots of the empirical logits are examined to explore potential relationships between the probability of duckling survival and both brood hatch date and number of ponds surrounding a nest. As we expect and mention earlier, duckling

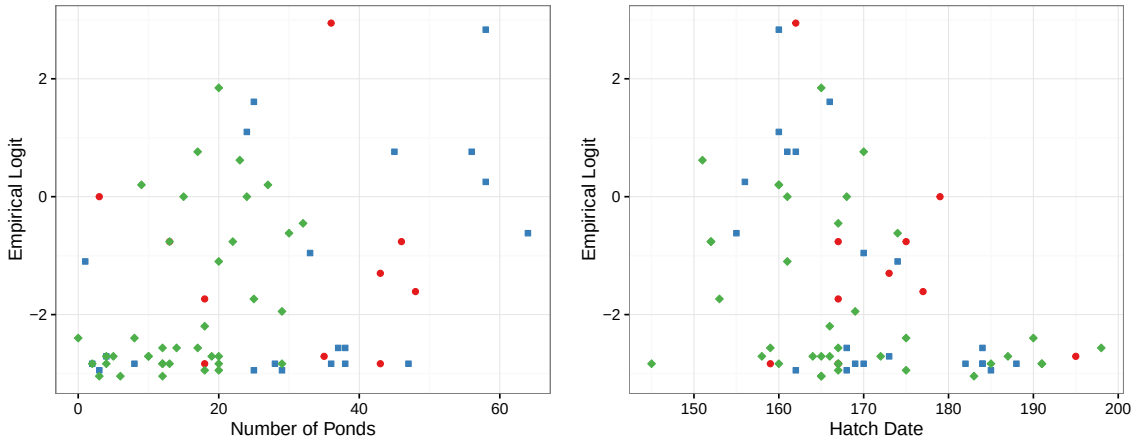

Figure 3: Plots of empirical logits versus the explanatory variables of interest. The left panel shows a pattern of increasing survival probabilities as the number of ponds increases. The right panel suggests a quadratic term for hatch date could describe the pattern between this variable and duckling survival probability. Points are coded by year - red circles for 1987, blue squares for 1988, and green diamonds for 1989.

survival probability on the logit scale appears to increase as the number of ponds near a brood's hatch site increases (Figure 3, left panel). For brood hatch date, duckling survival probability on the logit scale appears to show a pattern that initially increases as hatch date increases and then decreases for later hatch dates (Figure 3, right panel). This suggests that there could be an optimal hatch date or time period associated with high duckling survival and that hatch dates outside this zone, either earlier or later, are associated with lower duckling survival.

After a Julian date of approximately 180, all of the broods have zero surviving ducklings (Figure 3, right panel), suggesting that later hatch dates may be associated with much lower survival probabilities than earlier hatch dates in these data. This will be an important aspect of the data that will need to be captured by the model in order to make reasonable conclusions about the relationship between the probability of duckling survival and hatch date. Additionally, in order to understand how the number of ponds is related to duckling survival probabilities, it is also important to adequately account for how duckling survival probabilities are related to hatch date. A quadratic term for hatch date in the logistic regression model may be able to capture this apparent pattern of initially increasing and then decreasing probability of duckling survival as hatch date increases. The quadratic term for hatch date seems reasonable even though the decrease in survival for later hatch dates may be more severe than would be expected if the relationship was truly quadratic.

Next, the relationship between duckling survival and these covariates is examined with additional empirical logit plots. For these plots, we again display the empirical logits versus a covariate, but now examine observations separately based their values of the other explanatory variable. Here we are thinking about conditioning on values of one of the variables and then examining the relationship between duckling survival probabilities and the other variable. The plots showing the empirical logits versus number of ponds with broods separated by hatch date suggests that the relationship

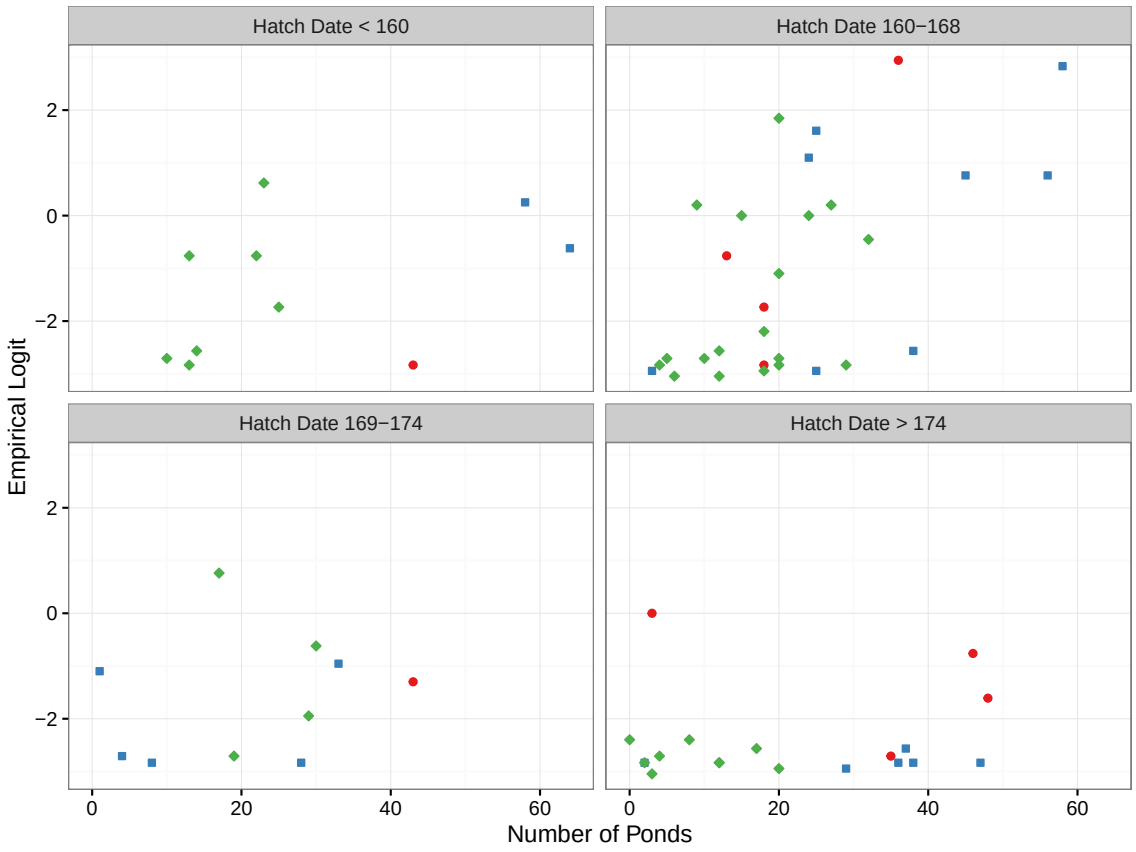

Figure 4: Plots of empirical logits versus number of ponds for broods with different hatch dates. The panels (starting in the top left) show the relationship for broods with hatch dates before Julian date 160, between 160 and 168, between 169 and 174, and for dates after 174. Points are coded by year - red circles for 1987, blue squares for 1988, and green diamonds for 1989.

between duckling survival probabilities and the number of ponds variable changes depending on the hatch date of the brood (Figure 4). The ranges of hatch dates we use to separate broods roughly reflect the pattern seen in the empirical logit plot versus this variable (Figure 3). The characterization of hatch date by these categories is done only for the purposes of creating exploratory plots, and not in the models where it is treated as a continuous variable. This pattern may be difficult to assess with these plots alone, but if the relationship between duckling survival probabilities and number of ponds changes depending on hatch date, an interaction term between hatch date and number of ponds may be needed when fitting the logistic regression

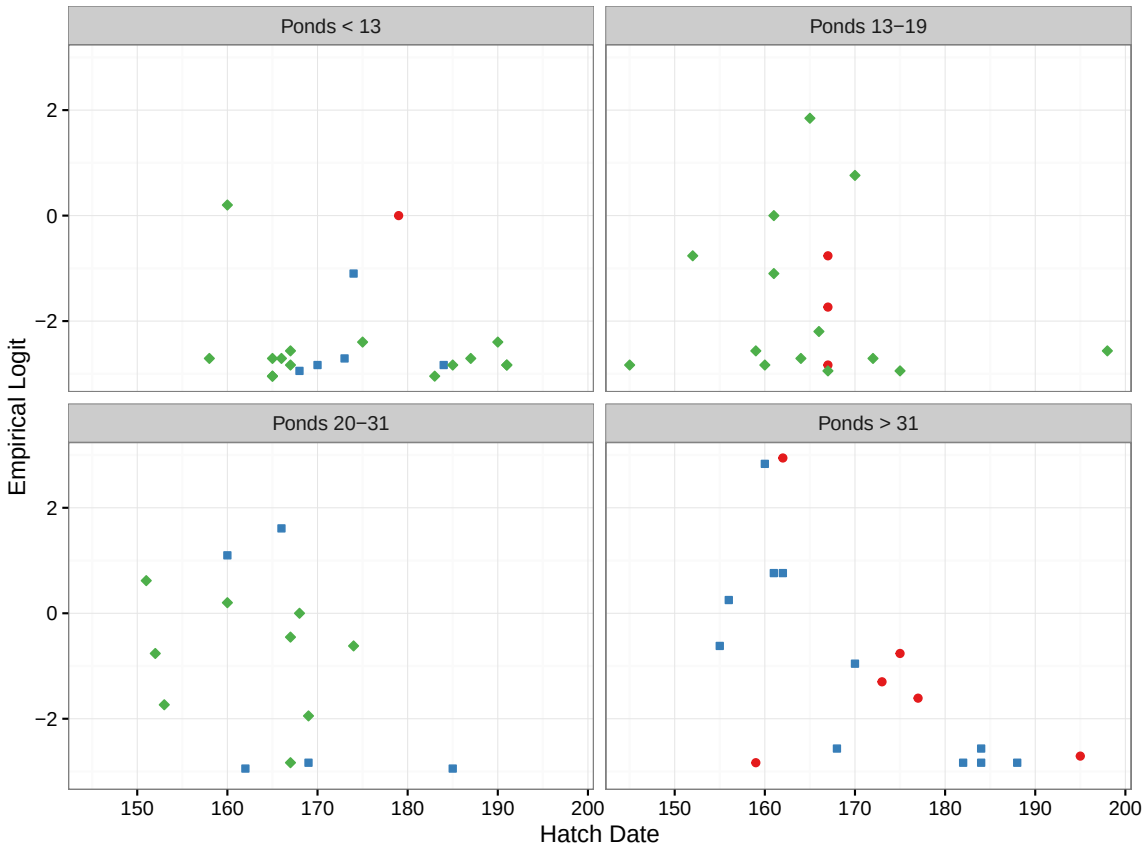

Figure 5: Plots of empirical logits versus hatch date for broods with different numbers of ponds. The panels (starting in the top left) show the relationship for broods with number of ponds less than 13, between 13 and 19, between 20 and 31, and for dates after 31. Points are coded by year - red circles for 1987, blue squares for 1988, and green diamonds for 1989.

model to these data. The plots of the empirical logits versus hatch date with broods separated by number of ponds also suggests an interaction between these variables may be needed (Figure 5). Here, the evidence of an optimal hatch period appears more apparent for broods with a large number of ponds than for broods with fewer ponds, but again these patterns are difficult to assess based on these plots alone. However, an interaction between these variables does make sense ecologically. If the ducklings from a brood hatched at a poor time where food availability is low or predator presence is high, then being in an area of high quality habitat might not be very beneficial. Or, thinking about this interaction in the other direction, ducklings

that hatch in an area with poor habitat quality may have a lower probability of survival regardless of when the eggs hatch.

It is important to note some of the difficulties in assessing the empirical logit plots for these data. While a sample size of 69 total broods may not appear small, these plots are less informative in this case because so many of the observed counts for these broods are quite small. Considering that 40 of the 69 broods had zero surviving ducklings after 30 days, we expect that these plots may not adequately illustrate the patterns that exist between duckling survival probabilities and these covariates. Additionally, we are interested in how duckling survival probabilities are related to both a brood's hatch date and the number of ponds surrounding the nest site. Making inference for survival across a wide range of values for both of these covariates will be difficult because we observe small sample sizes for some particular combinations of these variables. Again, this issue is exacerbated by the large number of broods with zero surviving ducklings. Both of these characteristics of these data make assessing the empirical logit plots difficult. We should also expect these issues to lead to some difficulties when fitting the logistic regression model.

Despite these limitations, we use the patterns in the exploratory plots to inform the logistic regression model structure for analyzing these data. We also base these models on what structure seems reasonable from an ecological standpoint as discussed above. Using all of this information, it seems reasonable that both hatch date and number of ponds explain some of the heterogeneity in duckling survival probabilities. For hatch date, the pattern seen in the empirical logits may be captured using a quadratic term for this variable. Interactions between hatch date and the number of ponds make sense ecologically and may be suggested in some of the empirical logit plots (Figures 4 and 5). The potential for hatch date and habitat conditions to have interacting impacts on duckling survival has been previously speculated (Pearse and Ratti 2004).

4 Analysis

4.1 Maximum Likelihood Approach

The structure of the first logistic regression model fit to these data is guided by the empirical logit plots and the ecological knowledge we discuss in Section 3. Year is not included in the model based on the assumption that the number of ponds variable is accounting for all differences in the probability of duckling survival due to differing habitat conditions across these years. A ML approach using the `glm()` function in R is initially used to obtain estimates for the coefficients in the following model:

$$Y_i \sim \text{Binomial}(m_i, p_i), \tag{4.1}$$

$$\text{logit}(p_i) = \beta_0 + \beta_1 \text{NP}_i + \beta_2 \text{HD}_i + \beta_3 \text{HD}_i^2 + \beta_4(\text{NP}_i \cdot \text{HD}_i) + \beta_5(\text{NP}_i \cdot \text{HD}_i^2), \tag{4.2}$$

where Y_i is the number of surviving ducklings for brood i, m_i is the number of eggs that hatch in the brood, and p_i is the probability of duckling survival for that brood. The logit link models each p_i using hatch date (HD), squared hatch date (HD^2), number of ponds (NP), an interaction between number of ponds and hatch date (NP·HD) and an interaction between number of ponds and the quadratic term of hatch date (NP·HD^2). These are all brood-level explanatory variables. We standardize (center and scale) the hatch date and number of ponds variables before performing this analysis in order to be able to compare results to the Bayesian models fit in the following section (4.2).

Table 2: Coefficient summary of ML estimates using the glm function in R

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-1.2673	0.1500	-8.45	0.0000
nponds	0.8383	0.1877	4.47	0.0000
hatchd	-1.3726	0.2725	-5.04	0.0000
hdsq	-0.8413	0.2077	-4.05	0.0001
nponds:hatchd	-0.5066	0.2131	-2.38	0.0174
nponds:hdsq	-0.7121	0.2292	-3.11	0.0019

The estimates and associated standard errors indicate strong evidence that all of the parameters in this model differ from zero (Table 2). We do not show plots of duckling survival probability estimates from this model, but instead display the results of the same model fit using a Bayesian approach in the following section. The ML approach and Bayesian approach using vague priors yield similar inferences for this model.

4.2 Bayesian Logistic Regression

4.2.1 Model

The same model structure from Equations 4.1 and 4.2 can also be fit using a Bayesian approach. The Bayesian version of this model also includes a prior distribution for all model parameters. In this case, we specify independent, normal prior distributions for each regression coefficient, that is, $\beta_k \sim \text{N}(0, 900)$ for $k = 0, \ldots, 5$. This prior is considered fairly vague because the large variance parameter places similar prior density on a large range of possible values these regression parameters can take on. A vague prior allows the data to dominate the resulting posterior distribution. If the improper Uniform$(-\infty, \infty)$ prior (not a true probability distribution) is placed on each parameter giving equal density to all possible values, the resulting posterior modes for each parameter will be equivalent to the ML estimates above.

This model is fit using MCMC with the `rstan` package and our code is available in the Appendix. The hatch date and number of ponds variables are centered and scaled for this analysis because it speeds convergence considerably in this case. Four chains of 2000 iterations are used, each with random initial values automatically chosen by Stan, and the first half of each chain is discarded as a warmup. Since all $\hat{R}$ values are less than 1.01 and the traceplots for every model parameter (see Appendix Figure 40) show good mixing, it appears the model has converged sufficiently. Consistent with

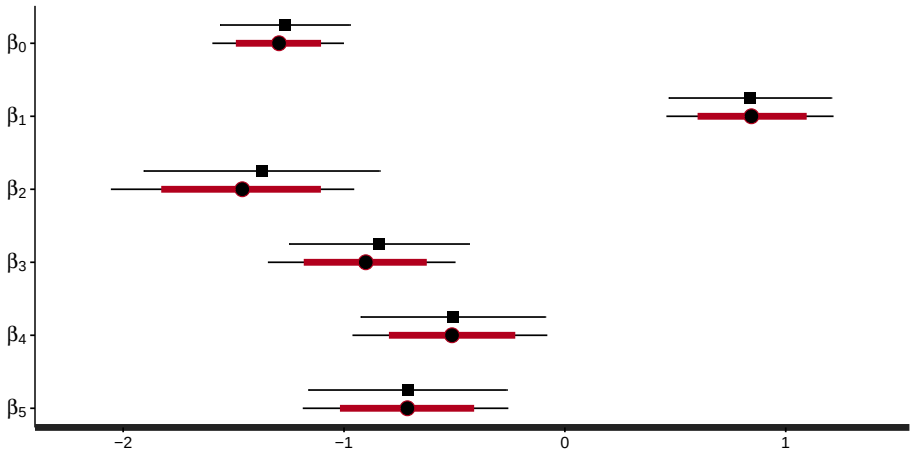

Figure 6: Summary of the posterior draws for each parameter from the initial Bayesian model. The posterior mean for each parameter is shown by the point (circle) along with the 80% PIs (thick, red) and 95% PIs (thin, black) shown by lines. For comparison, the corresponding ML estimates (squares) and 95% Wald's based CI (lines) are shown slightly above each Bayesian parameter summary.

using a vague prior, the posterior means and 95% PIs for each parameter are similar to the corresponding ML point estimates and 95% Wald's based CIs (Figure 6).

4.2.2 Results

We can use the posterior draws to construct posterior surfaces of the probability of duckling survival based on different combinations of hatch date and number of ponds. Using the regression coefficients from each saved iteration of the MCMC algorithm, the probability of duckling survival is calculated over a grid for every combination of hatch date and number of ponds values covering the range of each variable in the observed data. These calculations use the posterior draws of the regression coefficients to approximate the posterior distributions for the probabilities of duckling survival at these combinations of hatch date and number of ponds. The posterior draws for these derived parameters can then be summarized using the mean, median, or various quantiles to describe and visualize these posterior distributions.

The duckling survival probability surface created using the posterior mean at each

combination of the covariates shows that survival increases as hatch date increases, up to a point and then drops off substantially, with the initial increase in survival probability being more substantial for many ponds compared to a lower number of ponds (Figure 7). Stated a different way, the increase in the mean posterior survival probability associated with increasing the number of ponds is less pronounced when hatch date is not at a moderate value (around 160). This is describing the interaction between the hatch date and number of ponds variables. The mean posterior survival probability surface shows very small (≈ 0) survival probabilities for ducklings from broods with a hatch date later than 180 which is consistent with the observed data.

Duckling survival probability surfaces created using the 2.5% and 97.5% quantiles of the posterior distributions both have a pattern similar to that in the mean posterior surface (Figure 8). One aspect to note is the 'folding' of the corners of the upper

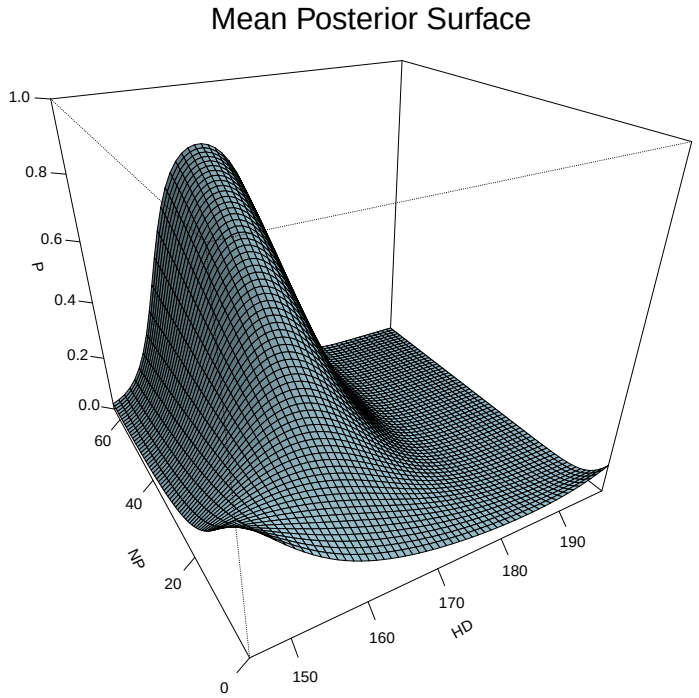

Figure 7: Based on the initial Bayesian logistic regression model, the mean posterior duckling survival probability surface for different combinations of hatch date and number of ponds.

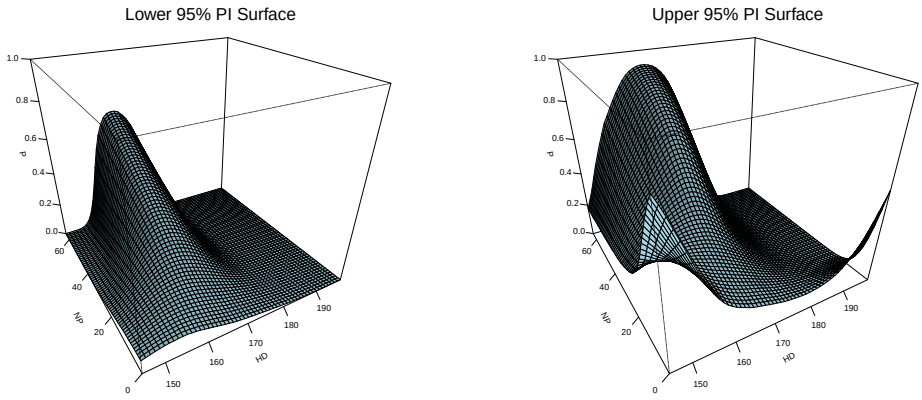

Figure 8: Based on the initial Bayesian logistic regression model, the 95% PI bounds for the duckling survival probability surface for different combinations of hatch date and number of ponds.

95% PI surface for low number of ponds and either very early or very late hatch dates. The large uncertainty in the probability for these regions is a result of little information available for these combinations of hatch date and number of ponds. The little available information is due both to the low sample sizes in these regions as well as the vague prior for the regression parameters in this model.

4.2.3 Posterior Predictive Checks

This model can be assessed using posterior predictive checks to evaluate how consistent predictions from the model are with the observed data. For each iteration of the MCMC algorithm, a posterior predictive dataset of surviving duckling counts (y^{rep}) is simulated based on the survival probabilities for each brood associated with that iteration and the actual number of ducklings in each brood from the observed dataset. In total, 4000 posterior predictive datasets are created and compared to the observed data in the following assessments. These posterior predictive datasets are considered potential realizations of duckling counts which could be observed under the fitted model. Various summary statistics can be calculated for each posterior predictive dataset and then compared to the value of the statistic using the observed

duckling counts. The comparisons of posterior predictive distributions to observed statistics can be used to compare aspects of the fitted model, and counts it could generate, to the observed dataset. Posterior predictive checks consistent with the observed data illustrate aspects of the data captured adequately by the model. Conversely, if an observed statistic is unusual compared to the posterior predictive datasets, that posterior predictive check suggests the model is not adequately capturing that aspect of the observed data. In other words, it indicates the model needs to be refined in order to make the model more consistent with the observed data.

The first statistic we use as a posterior predictive check is the average proportion of surviving ducklings for broods with different combinations of hatch date and number of ponds. For number of ponds, two classes are examined, broods with

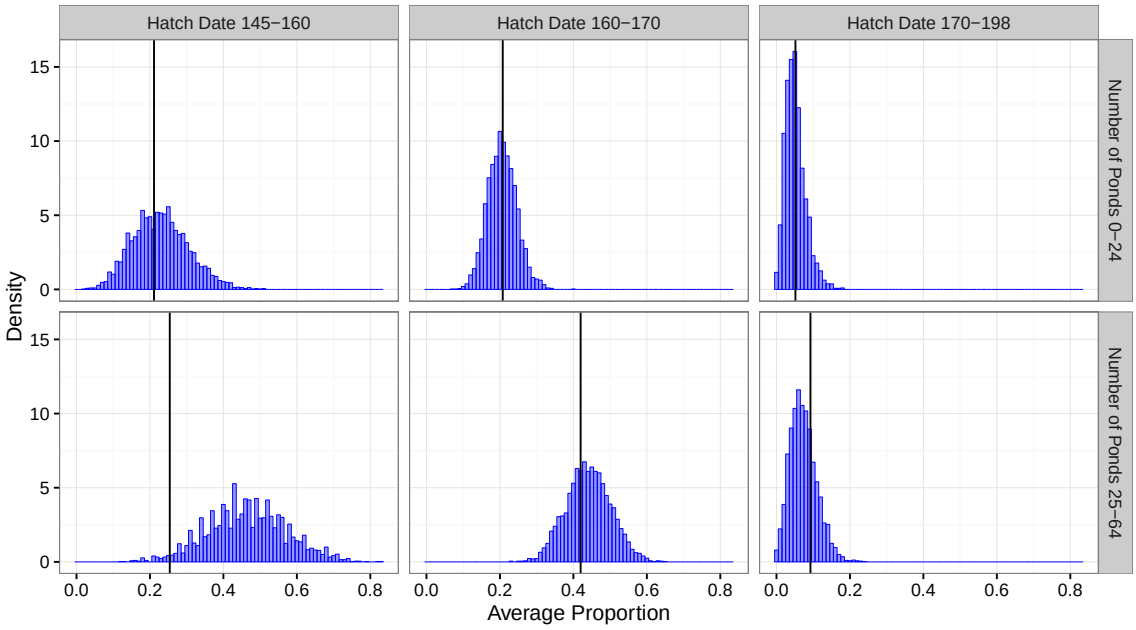

Figure 9: Posterior predictive checks for the average proportion of surviving ducklings for different classes of number of ponds and hatch dates. The three columns show broods with a hatch date before 160, from 160 to 170, and later than 170. The first row shows broods with less than 25 ponds and the bottom row shows broods with at least 25 ponds. Densities are shown for each statistic using the posterior predictive datasets and the solid line shows the value using the observed data.

less than 25 ponds in the nest vicinity and those with at least 25 ponds. Hatch date is divided into three classes - before Julian date 160, between 160 and 170, and after date 170. Each combination of the classes for the two variables are examined giving a total of six statistics. Except for broods with early hatch dates (before 160) and many ponds (at least 25), the observed average proportions of surviving ducklings from these classes fall near the center of their posterior predictive distributions (Figure 9). For the broods with the early hatch dates and many ponds, only 2.35% of the distribution falls below the observed mean proportion. However, since there are only four broods in this category and one has a count of zero, this check may not be a strong indication of a discrepancy between the model and observed data. For instance, if the brood with a zero count is excluded from the calculation, the observed average proportion is much closer to the center of the posterior predictive distribution. Overall, this check suggests that the model is adequately describing the average proportion of surviving ducklings across the different combinations of hatch date and number of ponds.

The posterior predictive check using the standard deviations of the proportions of surviving ducklings from these same categories suggests that the model is not adequately describing the variability in the proportions of ducklings (Figure 10). Many of the observed statistics are much larger than their corresponding posterior predictive distributions. In particular, for categories with higher estimated survival probabilities, the observed variability in the proportions is much higher than that for the posterior predictive datasets. This check indicates lack of agreement between the data and the model based on the standard deviations of proportions in these categories. In particular, it indicates that there is more variability in the observed proportions than is expected for this model. After accounting for heterogeneity in survival probabilities due to the covariates, variability in the counts from the model is due to them being random variables following a binomial distribution. Evidence of additional variation in these data is not surprising, and illustrates an aspect of these

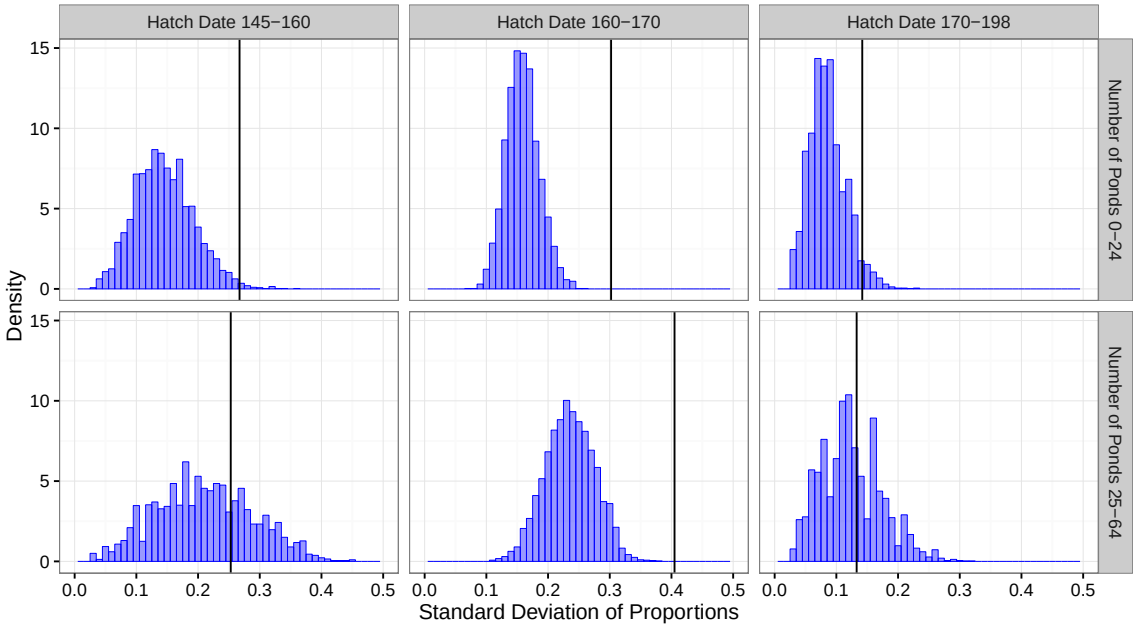

Figure 10: Posterior predictive checks for the standard deviations of the proportions of surviving ducklings for different classes of hatch date and number of ponds. The three columns show broods with a hatch date before 160, from 160 to 170, and later than 170. The first row shows broods with less than 25 ponds and the bottom row shows broods with at least 25 ponds. Densities are shown for each statistic using the posterior predictive datasets with the solid line showing the true observed value.

data not being captured by the model.

More variation than is expected if the counts are binomially distributed is known as 'extra binomial variation', or more generally as 'overdispersion'. This can be due to a variety of reasons. For these data, genetic variability can result in additional variability in the duckling survival probabilities within a brood. Furthermore, additional brood characteristics that are not accounted for with the number of ponds and hatch date variables can introduce additional heterogeneity in the survival probabilities. For instance, the mother's age or brood rearing experience can impact the probability of survival for ducklings in her brood, but this variable is not available in these data. These scenarios will result in either a lack of independence among observations or additional variation in survival probabilities which both can result in overdispersion.

More observed counts of zero than is expected from the model can lead to overdispersion as well. To investigate how the observed data compare to the model using this characteristic, posterior predictive checks are examined for the number of broods with zero surviving ducklings. We also look at a posterior predictive check for the number of broods with all ducklings surviving. The observed number of broods with zero surviving ducklings is much larger than any of the counts from the posterior predictive datasets and the observed number of broods with all ducklings surviving is slightly larger than the corresponding posterior predictive distribution (Figure 11). These checks suggest that this model is not adequately describing these characteristics of the observed data. These checks may also indicate the data are showing more variability than would be expected if the counts of surviving ducklings are binomially distributed.

All broods with a hatch date on or after a Julian date of 180 have zero surviving ducklings, and a large number of zero counts from the hatch dates before this are also present (Figure 3). We can further examine the zero counts in the posterior predictive datasets based on whether they are for broods before or after a hatch date of 180

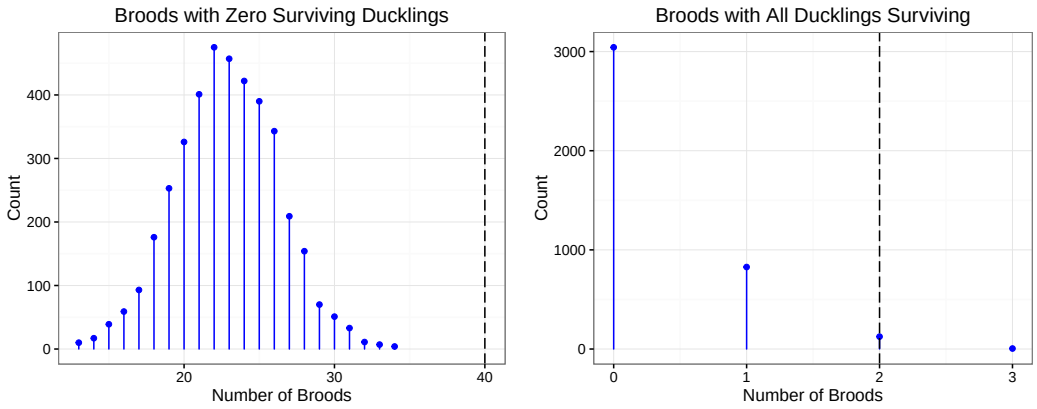

Figure 11: Posterior predictive checks for the number of broods with zero surviving ducklings (left panel) and all surviving ducklings (right panel). In each, the horizontal line shows the corresponding statistic using the observed data. Having 40 broods where all ducklings died is extremely unusual under the fitted model. The observed number of broods with all ducklings surviving is also unusual based on this model.

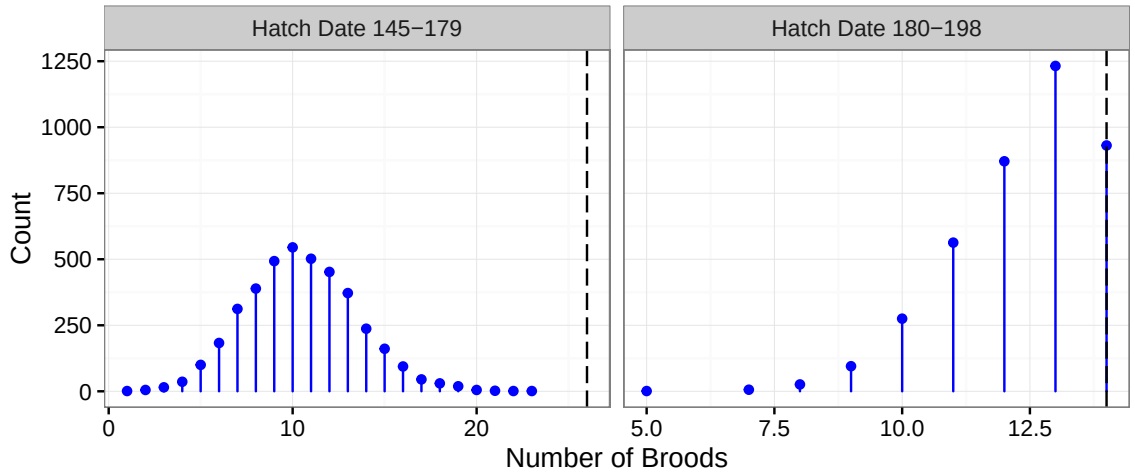

Figure 12: Posterior predictive check for the number of broods with zero surviving ducklings split by hatch date. The left panel shows earlier hatch dates (145-179) and the right panel shows later hatch dates (180-198). Based on this cutoff, all ducklings died in broods for the later hatch dates. In each, the observed number of broods with zero counts is shown by the vertical dashed line.

and compare these to the corresponding counts from the observed data. This final posterior predictive check examines the issue of the zero counts more thoroughly. This assessment indicates the structure of this model appears to adequately capture that all ducklings die after hatch date 180 (Figure 12, right panel). Therefore, the covariate structure for the probability of duckling survival (Equation 4.2) is accounting for at least some of the observed zero counts. However, this assessment also illustrates that for the broods hatching earlier, the number of zero counts in these data is much larger than the posterior predictive distribution of this summary measure (Figure 12, left panel).

4.3 Accounting for Overdispersion using Normal Errors

4.3.1 Model

The posterior predictive checks (Section 4.2.3) identify aspects of the observed data that are inconsistent with the fitted model. In particular, there is evidence

of more variability in the responses than explained by the model. This could be due, at least in part, to the large number of broods with zero surviving ducklings in comparison to what is expected under the model. While including the quadratic term for hatch date in the model captures that all broods with a hatch date later than 180 have zero surviving ducklings, it is not adequately describing the large number of zero counts before this cutoff. To account for the additional variability in these data, we explore incorporating overdispersion by adding a normally distributed error term for each brood (ϵ_i). This is one way that overdispersion can be added to GLMs and other approaches are described in the following sections. The overdispersion model using normally distributed errors can be expressed notationally as:

$$Y_i \sim \text{Binomial}(m_i, p_i), \tag{4.3}$$

$$\text{logit}(p_i) = \beta_0 + \beta_1 \text{NP}_i + \beta_2 \text{HD}_i + \beta_3 \text{HD}_i^2 + \beta_4 (\text{NP}_i \cdot \text{HD}_i) + \beta_5 (\text{NP}_i \cdot \text{HD}_i^2) + \epsilon_i, \tag{4.4}$$

$$\epsilon_i \sim \text{N}(0, \sigma^2), \tag{4.5}$$

$$\beta_k \sim \text{N}(0, 900) \text{ for } k = 0, \ldots, 5, \tag{4.6}$$

$$\sigma \sim \text{half-Cauchy}(0, 2.5). \tag{4.7}$$

For this model, the counts are still assumed to be binomially distributed (Equation 4.3), but now a normally distributed error term is added to the logit of the probability of survival for each brood (Equations 4.4 and 4.5). In the previous model, if two broods have the same hatch date and number of ponds, then they have the same probability of duckling survival. Any differences in the counts for these broods is assumed to be due to variability in the binomial process. The error terms in this model, however, allow broods with the same covariate pattern to have different probabilities of survival. This results in the binomial counts having additional variability in comparison to before and is one way that overdispersion can

be accounted for in a model.

The same vague prior as in the previous model is placed on the regression parameters (Equation 4.6). The prior we assign to the standard deviation of the distribution of the error terms (Equation 4.7) is meant to be weakly informative. It includes enough information to provide realistic estimates because it places more density on plausible values, but is vague enough that it does not overly influence the posterior distribution. This prior has most of its density on values of σ close to zero and almost no density on values greater than ten. We expect this parameter to take on small values since survival probabilities are modeled on the logit scale with centered and scaled covariates. For this reason, it makes sense that the distribution of the ϵ_i values would have a variance much less than 100.

The posterior distribution for σ appears quite high and results in the uncertainty in the regression parameters being much larger than that from the previous model (Figure 13). There is no longer strong evidence that all of the regression parameters differ from zero when this additional variability is accounted for in the model. As with the previous model, the posterior draws can be used to create surfaces describing

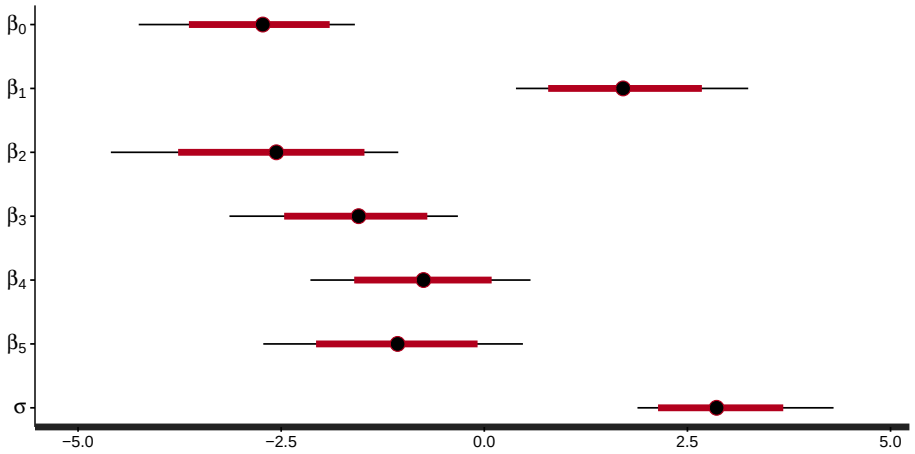

Figure 13: Summary of the posterior draws for each parameter from the Bayesian model with overdispersion using normal errors. The posterior mean for each parameter is shown by the point (circle) along with the 80% PIs (thick, red) and 95% PIs (thin, black) shown by lines.

the posterior distributions of duckling survival probability over different hatch dates and number of ponds. Again, these plots can be used to provide inference for the relationships of duckling survival with the number of ponds and hatch date variables based on this fitted model and are easier to understand than examining the individual coefficients.

4.3.2 Results

In comparison to the previous model (Section 4.2.2), the mean posterior survival probability surface does not appear to have changed substantially (Figure 14). This is not surprising - for instance, when including overdispersion in `glm()` using the 'quasibinomial' family, the estimates do not change but the standard errors are inflated. Using the mean posterior surface, inferences about the probability of duckling survival remain approximately the same. The mean probability of duckling survival increases as number of ponds increases with a more dramatic increase for broods with moderate hatch dates.

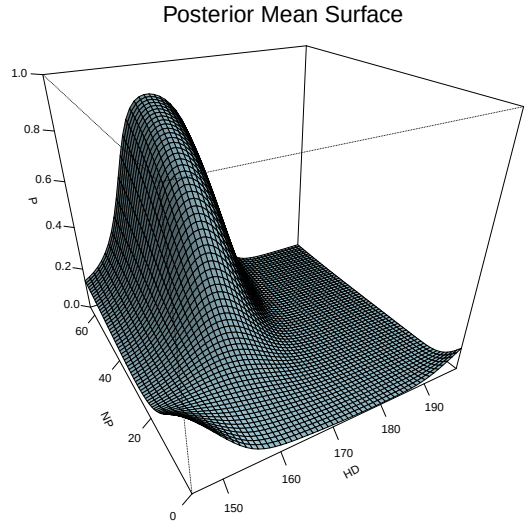

Figure 14: The mean posterior duckling survival probability surface for different combinations of hatch date and number of ponds based on the Bayesian model including overdispersion using normal errors.

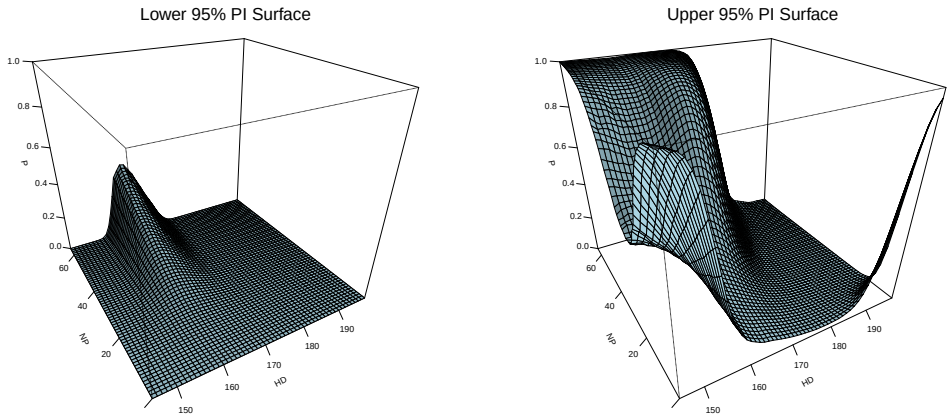

Figure 15: Based on the Bayesian model including overdispersion with normal errors, the 95% posterior interval duckling survival probability surface for different combinations of hatch date and number of ponds.

However, there is a considerable increase in the uncertainty associated with the survival probability surface in comparison to the previous model (Figure 15). Based on this model and the uncertainty for the parameters, there is not as clear of an interpretation for how duckling survival probabilities are related to hatch date and number of ponds. For instance, if we condition on a hatch date of 160 and think about how survival probability is associated with increasing the number of ponds, there could be a moderate increase (to around 0.4, lower bound Figure 15) or an extreme increase (to approximately 1.0, upper bound Figure 15). For earlier hatch dates, this uncertainty is even more extreme. Similarly, conditioning on number of ponds also shows a large amount of uncertainty and makes it difficult to draw any sort of reasonable conclusions regarding the association between duckling survival probabilities and these covariates. Essentially, the fit of this model suggests that there are many, and extremely different, plausible survival probability surfaces based on these data. This appears to be due to the large amount of heterogeneity in survival probabilities resulting from differences among broods (ϵ_i) that is not explainable by the hatch date or number of ponds variables.

4.3.3 Posterior Predictive Checks

We again use posterior predictive checks to compare this model to the observed data with the hope that the inclusion of overdispersion addressed the deficiencies of the previous model. The same posterior predictive checks from the previous model (Section 4.2.3) are examined. In these assessments, the posterior predictive datasets are considered for broods in this sample. This means that the ϵ_i parameter is taken into account for the survival probability associated with each brood when simulating duckling counts at each iteration. It is possible to consider out of sample broods when doing posterior predictive datasets where the error term is drawn from a distribution at each iteration, but we do not perform these assessments here.

Overall, these assessments now indicate more consistency between the observed

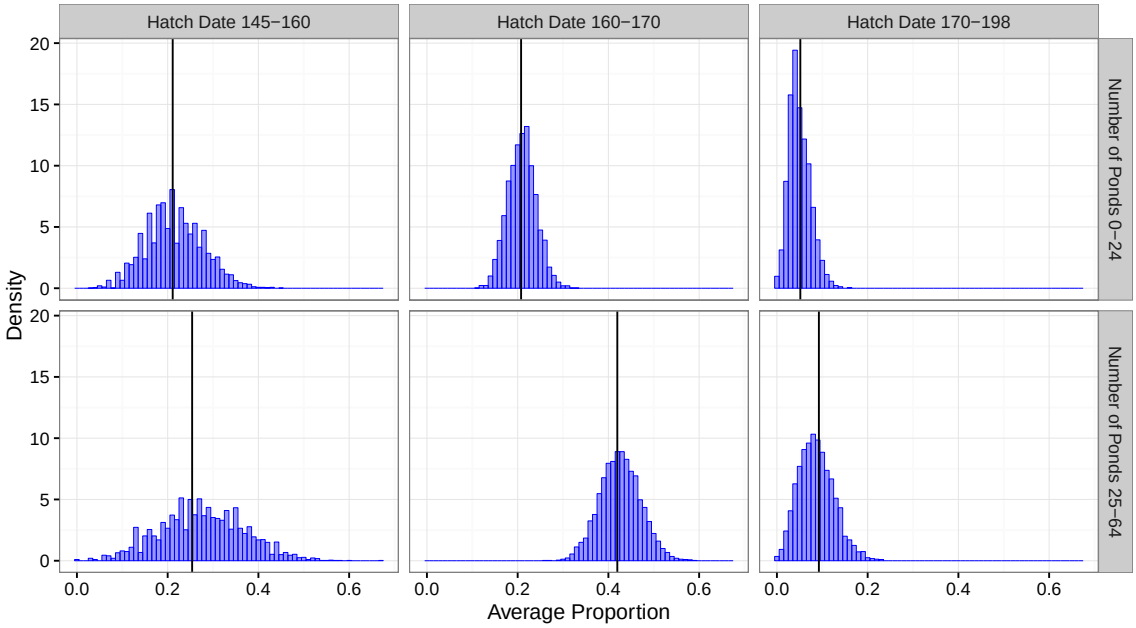

Figure 16: Posterior predictive checks for the model with overdispersion using normal errors on the average proportion of surviving ducklings for different classes of number of ponds and hatch dates. The three columns of plots show broods with a hatch date before 160, from 160 to 170, and later than 170. The first row shows broods with less than 25 ponds and the bottom row shows broods with at least 25 ponds. The densities are for the statistics from the posterior predictive datasets and the solid line shows the observed value.

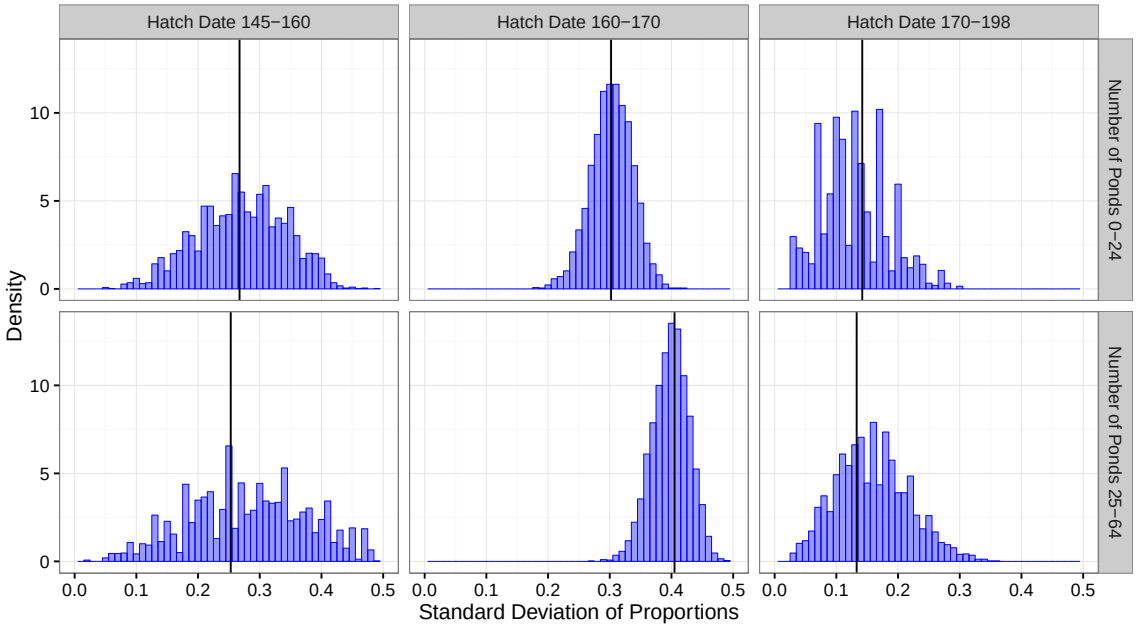

Figure 17: Posterior predictive checks for the standard deviation of the proportions of surviving ducklings for different classes of hatch date and number of ponds from the model with overdispersion using normal errors. The three columns of plots show broods with a hatch date before 160, from 160 to 170, and later than 170. The first row shows broods with less than 25 ponds and the bottom row shows broods with at least 25 ponds. Densities show the statistics for the posterior predictive datasets and the solid line shows the observed value.

data and the model using these summaries. Unsurprisingly, since the covariate structure of the model has remained the same, the observed mean proportions for different combinations of hatch date and number of ponds are still consistent with the corresponding posterior predictive distributions (Figure 16). More importantly, however, the inclusion of error terms in the model appears to add additional variability in the proportions from the posterior predictive datasets consistent with the observed proportions (Figure 17). This is expected and is the motivation for including the error terms in the model to begin with. Including overdispersion in the model also results in the observed number of broods with zero surviving ducklings or all surviving ducklings being consistent with the posterior predictive distributions for these quantities (Figure 18) which was not seen in the previous model. Additionally, the posterior predictive

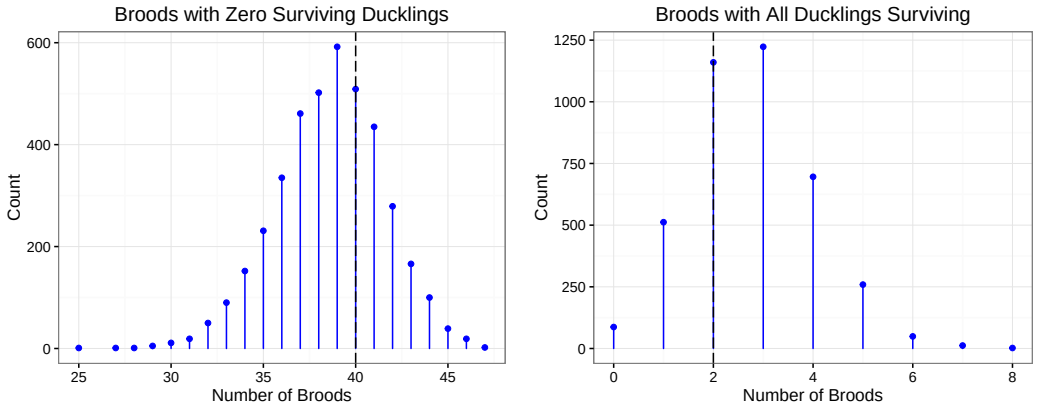

Figure 18: Posterior predictive checks for the number of broods with zero surviving ducklings (left panel) and all surviving ducklings (right panel) from the model with overdispersion. In each, the horizontal line shows the corresponding test statistic from the observed data.

distribution for the number of broods with zero surviving ducklings in the lower (145-179) hatch date range now includes the observed value for this count (Figure 19). As is desired, including additional variability in the probability of duckling survival using the normal errors is able to address both of the main deficiencies of the initial Bayesian model - the observed variability in proportions and the number of zero counts are now consistent with this model.

However, including overdispersion in this way adds so much variability in the duckling survival probabilities that there is substantial uncertainty associated with all of the model parameters. This model does not provide useful inferences about how duckling survival is related to hatch date and number of ponds because of the large amount of uncertainty in the 95% PI surface (Figure 15). Next, we explore additional models that could be used to analyze these data while still incorporating additional variability in the surviving duckling counts.

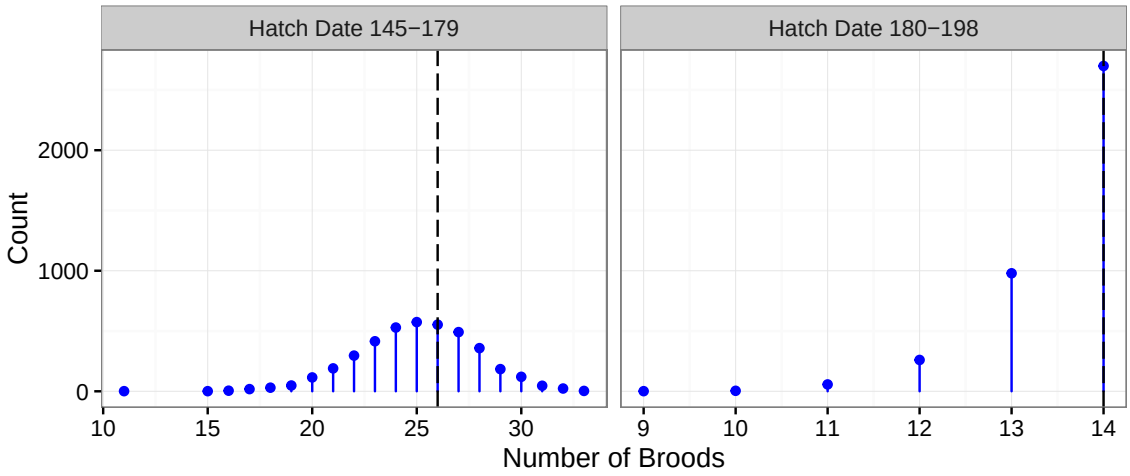

Figure 19: Posterior predictive check for the number of broods with zero surviving ducklings split by hatch date for the model with overdispersion. The left panel shows earlier hatch dates (145-179) and the right panel shows later hatch dates (180-198). Based on this cutoff, all ducklings died in broods for the later hatch dates. The observed number of broods with zero counts based on this distinction is shown by the vertical dashed line.

4.4 Beta-Binomial Model for Overdispersion

Adding normally distributed error terms to the logit of the modeled probabilities is frequently used to account for overdispersion in a Bayesian framework. However, as the description of the model above (Equations 4.3 to 4.7) shows, it makes a specific assumption about the structural form of the overdispersion. In that model, the errors (ϵ_i) are assumed to be independent, normally distributed random variables with a common variance. This may be appropriate in some scenarios, but there are additional ways to incorporate additional variability in the survival probabilities. For these data, the overdispersion appears to be quite large and results in very large uncertainty associated with the relationships of duckling survival probabilities with hatch date and number of ponds. Because understanding how survival probabilities relate to these covariates is the primary interest of this study, alternative solutions to account for the overdispersion in the observed counts will be explored here.

One alternative approach to do this is to use a beta-binomial logistic regression

model to analyze these data. Instead of assuming the logit of the survival probabilities are normally distributed, this model assumes the probabilities follow a beta distribution. This model is described as follows:

$$Y_i \sim \text{Binomial}(m_i, p_i)$$

$$p_i \sim \text{Beta}(\kappa\nu_i, \kappa(1-\nu_i))$$

$$\text{logit}(\nu_i) = \beta_0 + \beta_1 \text{NP}_i + \beta_2 \text{HD}_i + \beta_3 \text{HD}_i^2 + \beta_4(\text{NP}_i \cdot \text{HD}_i) + \beta_5(\text{NP}_i \cdot \text{HD}_i^2)$$

$$\beta_k \sim \text{N}(0, 900) \text{ for } k = 0, \ldots, 5,$$

$$\frac{1}{\kappa} \sim \text{half-Cauchy}(0, 5).$$

In this approach, the logit transformation uses the covariates to model the mean (ν) of the beta distribution for the probabilities of duckling survival. As in the previous model (Section 4.3.1), this allows broods with the same covariate pattern to still have different probabilities of survival. Again, there will now be additional variability in the duckling counts in comparison to the initial Bayesian model (Section 4.2) that did not incorporate any overdispersion. In this model, the precision (κ) of the beta distributions describing the survival probabilities is assumed constant regardless of the mean.

We do not show the results of this model here, but they are similar to those of the previous model with overdispersion. The posterior predictive checks again indicate that the variability in datasets produced by the model are consistent with the observed data, both in terms of the number of broods with zero surviving ducklings and standard deviations in proportions. However, there is again so much uncertainty in the duckling survival probability surface that no reasonable inferences are possible.

4.5 Zero-Inflated Binomial Approach

4.5.1 Model

The next model illustrates a different approach to incorporate more variability in the counts by incorporating an additional process that generates zeroes. The posterior predictive checks for the initial Bayesian logistic regression model (Section 4.2.3) suggest that directly addressing the zero counts may be a reasonable approach in the analysis of these data. This can be done using a zero-inflated model where it is assumed that there is an additional process that only results in zero counts. These zero counts are in addition to the zero counts from the binomial process of the model.

A zero-inflated model also makes sense ecologically due to the possibility that a 'catastrophic event' can result in an entire brood being killed at once. These catastrophic events can occur when a hen abandons her brood or when a predator kills all the ducklings in a brood at once. Predation does not, in theory, result in strictly zero surviving ducklings, but multiple studies on mallards indicate that predation events usually lead to the entire brood dying (Mauser, Jarvis, and Gilmer 1994; Pearse and Ratti 2004; Chouinard Jr, Arnold, and Haukos 2007). A zero-inflated model can be used to account for the additional zeroes that result from these catastrophic events. In this framework, a count of zero surviving ducklings can be observed if a brood experiences a catastrophic event *or* if a brood does not experience a catastrophic event but all of the ducklings die individually over the 30 days. Again, we use the logit link to model the probability of duckling survival, now conditional on the brood not experiencing a catastrophic event, with the hatch date and number of ponds covariates. We can also allow the probability of a brood not experiencing a catastrophic event to vary based on the available covariates, but will assume this probability is constant for this initial zero-inflated binomial model. This model is

described as follows:

$$\mathrm{P}(Y_i = y \mid p_i) = \begin{cases} (1-\pi) + \pi\left[(1-p_i)^{m_i}\right] & y = 0 \\ \pi\left[\binom{m_i}{y} p_i^y (1-p_i)^{m_i - y}\right] & 0 < y \leq m_i \\ 0 & \text{else} \end{cases} \tag{4.8}$$

$$\text{logit}(p_i) = \beta_0 + \beta_1 \text{NP}_i + \beta_2 \text{HD}_i + \beta_3 \text{HD}_i^2 + \beta_4(\text{NP}_i \cdot \text{HD}_i) + \beta_5(\text{NP}_i \cdot \text{HD}_i^2), \tag{4.9}$$

$$\text{logit}(\pi) = \alpha, \tag{4.10}$$

$$\beta_k \sim \text{N}(0, 900) \text{ for } k = 0, \ldots, 5, \tag{4.11}$$

$$\alpha \sim \text{N}(0, 900), \tag{4.12}$$

where π is the probability a brood does not experience a catastrophic event and is modeled on the logit scale by the α parameter. The remaining parameters of the model have the same definitions that are used in the initial logistic regression model (Section 4.2.1).

This model structure is slightly more complicated than the models presented in previous sections. The first line of the probability distribution for Y_i (Equation 4.8) shows how the additional zeroes in the counts are modeled. Catastrophic events occur with probability $(1 - \pi)$ and result in a zero count. Additionally, zeroes can be observed when no catastrophic event occurs but the binomial process generates a zero - this occurs with probability $\pi\left[(1-p_i)^{m_i}\right]$. This represents the zero-inflation process of the model. The second line of the specified probability distribution shows that counts greater than zero can occur from the binomial process when there is no catastrophic event. The probability of duckling survival is modeled using logistic regression equation (Equation 4.9) with the same structure from previous models. We

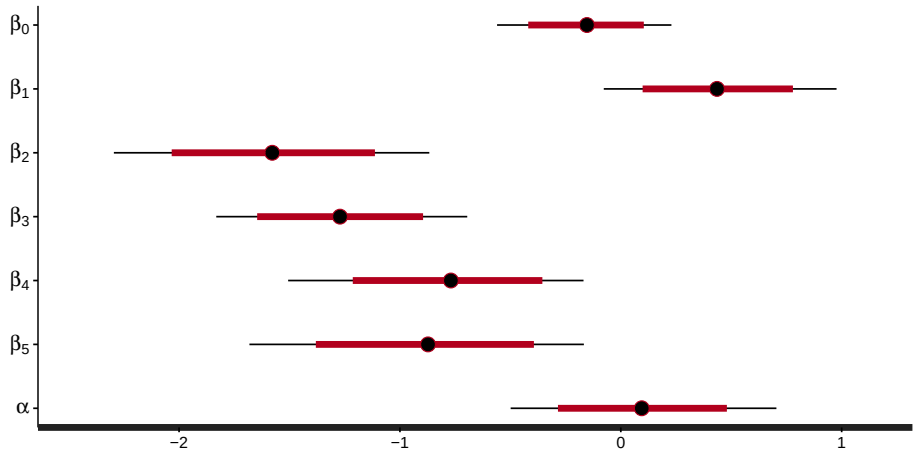

Figure 20: Summary of the posterior draws for each parameter from the initial zero-inflated model. The posterior mean for each parameter is shown by the point (circle) along with the 80% PIs (thick, red) and 95% PIs (thin, black) shown by lines.

use the logit link to model the probability of a brood not experiencing a catastrophic event (Equation 4.10), but initially will assume that this probability is constant. The same vague priors as we use for the previous models are specified for the regression parameters of this model (Equations 4.11 and 4.12).

4.5.2 Results

The regression coefficients associated with the probability of duckling survival (β parameters) are now conditional on the brood not experiencing a catastrophic event. Based on this model, the posterior mean probability a brood has no catastrophic event is 0.524 with an associated 95% PI from 0.378 to 0.669 (shown in Figure 20 on the logit scale). This interval appears to be quite large and we will consider including covariates to model this probability in the following section. Again, plots describing the posterior distribution for the duckling survival probability surface can be created using the posterior draws for the duckling survival regression parameters, but these probabilities are interpreted slightly differently because they are now conditional on no catastrophic event.

The surface of posterior mean duckling survival probabilities based on this model appears similar in overall shape to those from the previous models (Sections 4.2.2 and 4.3.2). Conditional on no catastrophic event, the probability of duckling survival increases as number of ponds increases for moderate hatch dates but is not beneficial for broods that hatch early or late (Figure 21). This same overall pattern is also seen after accounting for the uncertainty of these regression coefficients (Figure 22). There is still a lot of uncertainty in the conditional probability of ducklings survival for broods with a small number of ponds and either a very early or very late hatch date. Again, this is not surprising given the small amount of information available to inform the probability of survival for these covariate patterns. Overall, inferences from this model appear to be more precise than those from the earlier models including overdispersion. As a result, we obtain more reasonable and practical inferences regarding how duckling survival is related to these covariates. Posterior predictive checks will be used to confirm that this model adequately incorporates additional

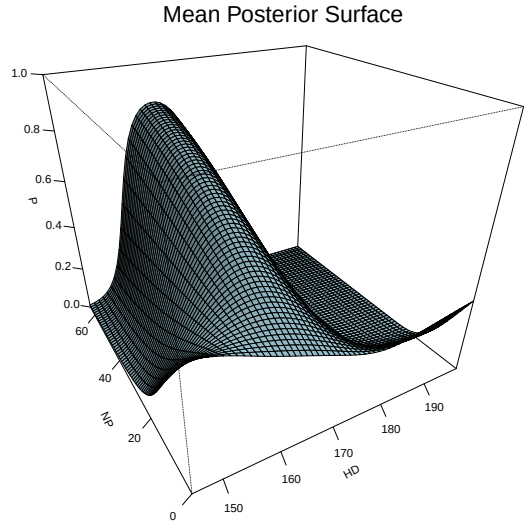

Figure 21: The mean posterior duckling survival probability surface for different combinations of hatch date and number of ponds based on the initial zero-inflated model. Note that this plot is now conditional on no catastrophic event killing the entire brood at once.

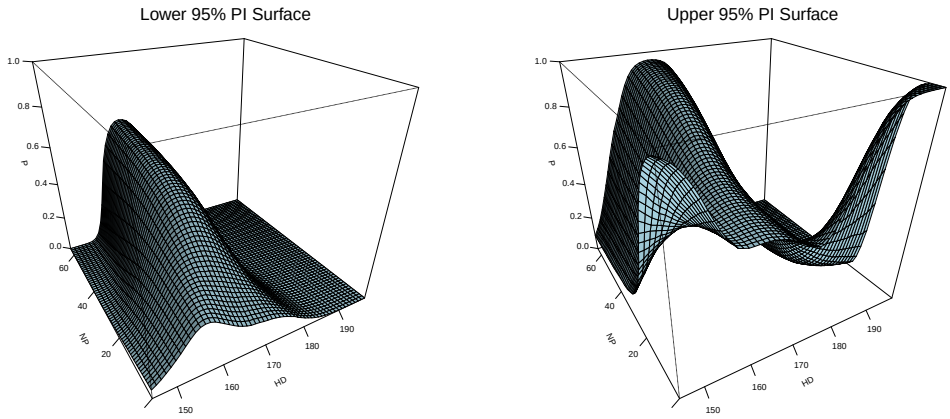

Figure 22: Based on the initial zero-inflated model, the 95% posterior interval duckling survival probability surface for different combinations of hatch date and number of ponds. Note that this plot is now conditional on the brood not experiencing a catastrophic event where all the ducklings are killed at once.

variability in the duckling counts through the zero-inflation process.

4.5.3 Posterior Predictive Checks

Here, we examine the posterior predictive checks from the previous sections (4.2.3 and 4.3.3) using this initial zero-inflated model. In this case, to simulate the posterior predictive datasets, for each saved iteration and brood in this dataset, a Bernoulli random variable is simulated using the probability of no catastrophic event from that iteration. Conditional on no catastrophic event, the number of surviving ducklings is simulated from a binomial distribution with m_i trials and probability p_i using the number of ponds and hatch date for that brood. If the Bernoulli random variable indicates a catastrophic event, zero ducklings survive for the brood in that simulation. Therefore, we consider the entire model including the zero-inflated process when generating these posterior predictive datasets.

The posterior predictive checks based on the average proportions of surviving ducklings are consistent with the observed data (Figure 23). For each of the categories we examine based on different combinations of hatch date and number of ponds, the

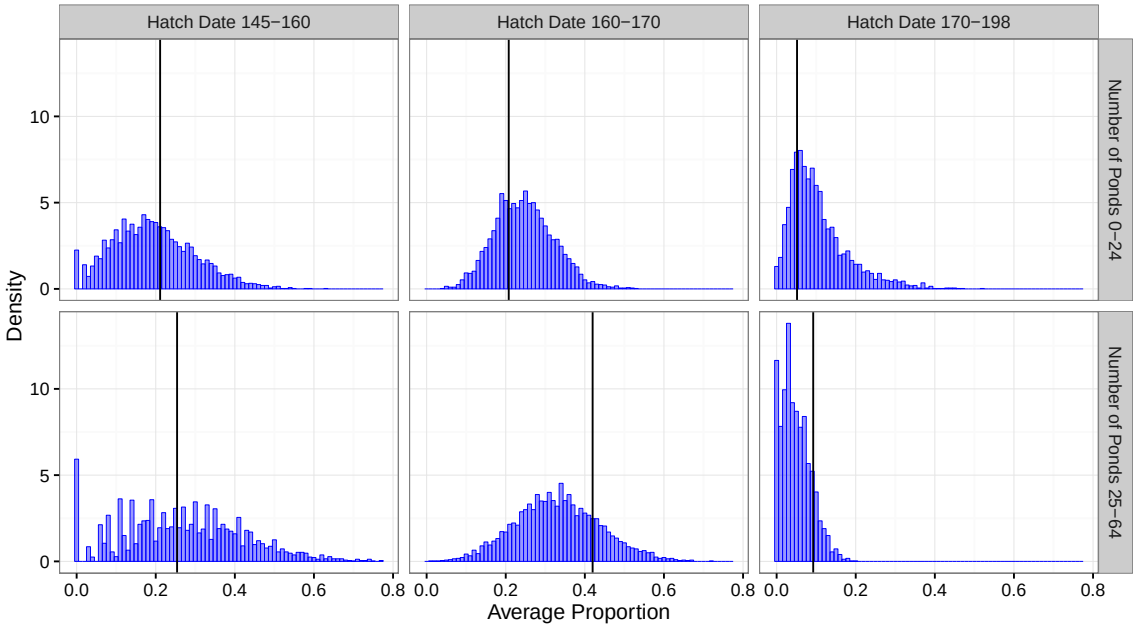

Figure 23: Posterior predictive checks of the zero-inflated model for the average proportion of surviving ducklings for different classes of number of ponds and hatch dates. The three columns of plots show broods with a hatch date before 160, from 160 to 170, and later than 170. The first row shows broods with less than 25 ponds and the bottom row shows broods with at least 25 ponds. The densities describe the posterior predictive distributions and the observed proportion is indicated by a vertical line in each plot.

average proportion of the observed surviving duckling counts are near the center of their respective posterior predictive distributions. For this check, the entire process of the model including the catastrophic event process and binomial process are being accounted for. The standard deviations of the proportions for broods in these same categories result in posterior predictive distributions that are also consistent with the standard deviations from the observed data (Figure 24). This assessment indicates that including a process that generates additional zeroes in the model is also a way to account for the overdispersion in the observed counts in comparison to what is expected if the counts are strictly from a binomial process.

Finally, the posterior predictive checks for the number of broods with zero surviving ducklings and with all ducklings surviving can be examined. Here,

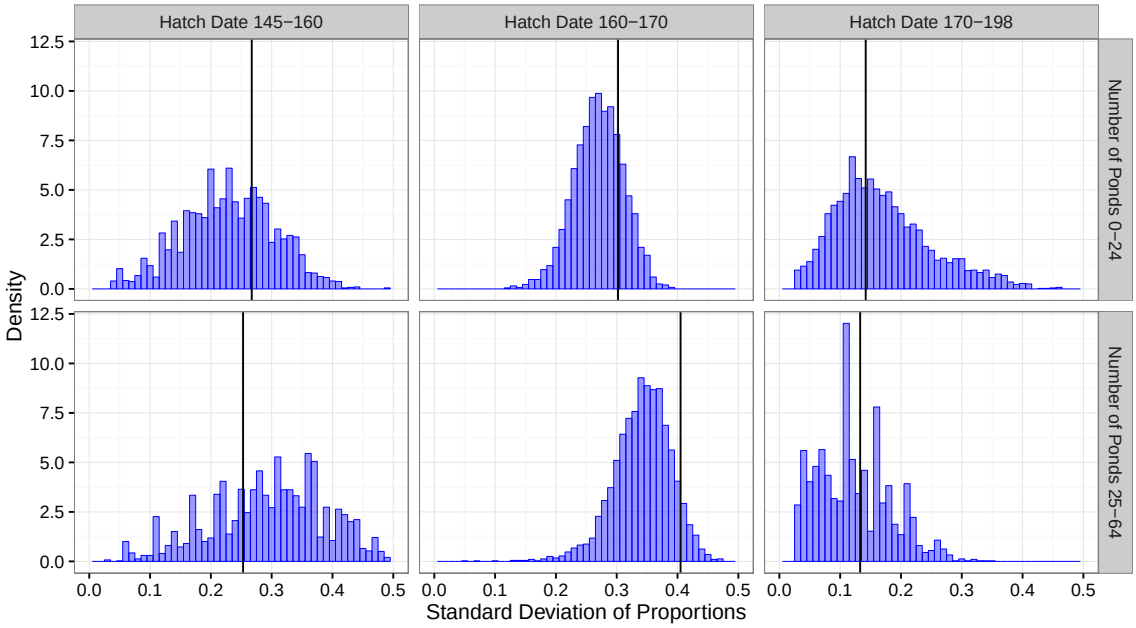

Figure 24: Posterior predictive checks for the standard deviation of the proportions of surviving ducklings for different classes of hatch date and number of ponds using the initial zero-inflated model. The three columns of plots show broods with a hatch date before 160, from 160 to 170, and later than 170. The first row shows broods with less than 25 ponds and the bottom row shows broods with at least 25 ponds. Plots of the density of the posterior predictive distributions and the observed statistic (vertical line).

the posterior predictive distribution for the number of broods with zero surviving ducklings is centered on the number of zeroes observed (40) in this dataset (Figure 25), and therefore illustrates the model is consistent with the observed data using this summary. This is not surprising considering the zero-inflated model specifically incorporates an additional process to generate zero counts. Observing two broods with every duckling surviving is still slightly unusual in comparison to the posterior predictive datasets from this model (Figure 25, right panel), but not so unusual that it suggests that the model is not adequately describing that aspect of these data.

The posterior predictive check based on the broods with zero surviving ducklings split by hatch date also shows that the posterior predictive datasets are consistent with the observed data for these summaries (Figure 26). Again, this is not

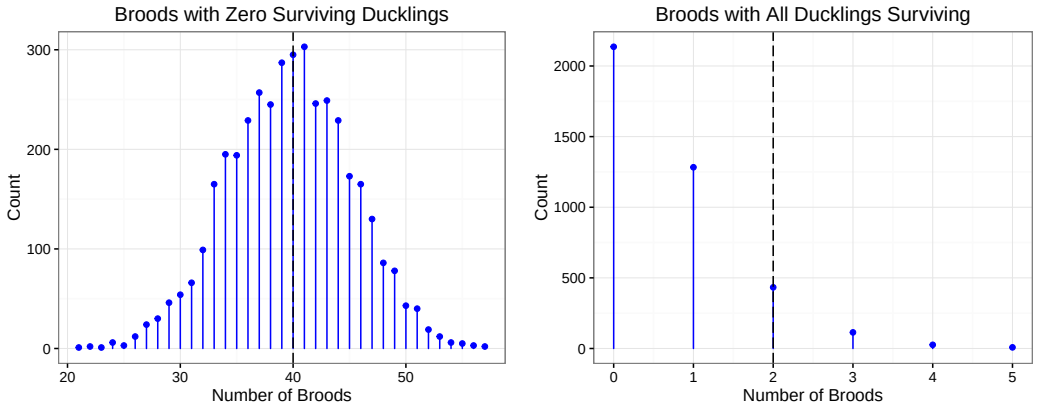

Figure 25: Posterior predictive checks for the number of broods with zero surviving ducklings (left panel) and all surviving ducklings (right panel) from the zero-inflated model. In each, the horizontal line shows the corresponding statistic from the observed data.

surprising considering that the zero counts are directly modeled. Overall, using these assessments, the model appears to be more in agreement with the observed data than the original binomial logistic regression model (Section 4.2.3) for these

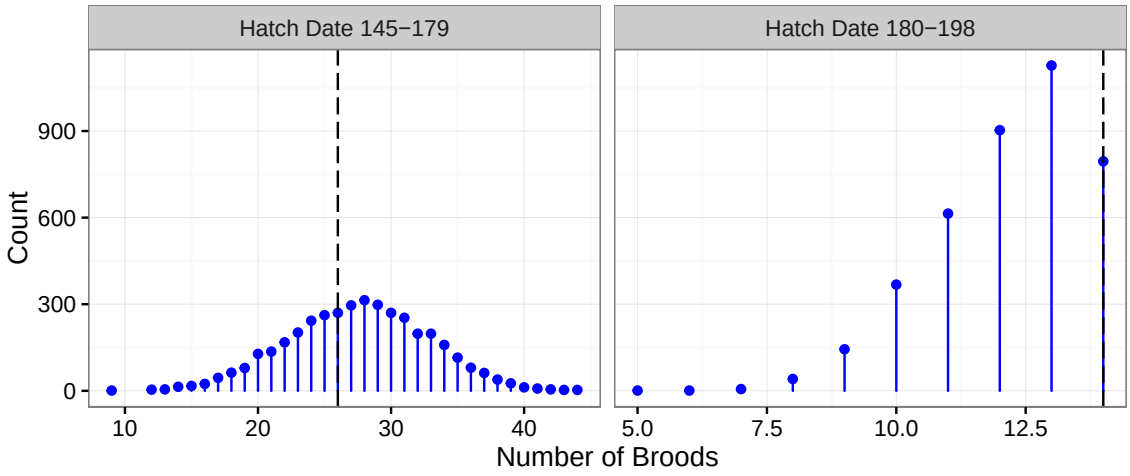

Figure 26: Posterior predictive check for the number of broods with zero surviving ducklings split by hatch date for the initial zero-inflated model. The left panel shows earlier hatch dates (145-179) and the right panel shows later hatch dates (180-198). Based on this cutoff, all ducklings died in broods for the later hatch dates. The observed number of broods with zero counts based on this distinction is shown by the vertical dashed lines in each plot.

data. This is also expected because the zero-inflated model is more complicated than the original binomial logistic regression model. This approach illustrates another way that overdispersion can be incorporated into a model for binomial counts. While none of these posterior predictive checks indicate inconsistencies between the data and the model with overdispersion using normal errors (Section 4.3.3), the zero-inflated model results in less uncertainty about the parameters of interest. Therefore, this model would be more useful for inferences about the probability of duckling survival and how it is related to the hatch date and number of ponds covariates.

4.5.4 Residual Plots

Residual plots provide an additional way to examine this model and how the observed and expected counts compare to one another. Examining these plots can also help diagnose potential misspecification of the model based on the available covariates. We start by examining the overall Pearson residuals, defined as

$$r_i = \frac{y_i - \mathrm{E}(Y_i)}{\sqrt{\mathrm{Var}(Y_i)}},$$

where $\mathrm{E}(Y_i) = m_i\pi_i p_i$ and $\mathrm{Var}(Y_i) = \pi_i m_i p_i(1 - p_i(1 - m_i(1 - \pi_i)))$. The expected value and variance of the counts from the zero-inflated model can be derived using the probability density function for Y_i (Equation 4.8). For a Bayesian model, residuals can be calculated at each iteration of the MCMC algorithm so multiple residual plots should be examined. Here, we show the overall Pearson residuals from three random iterations versus each covariate, but more iterations can be examined in practice. Examining these residuals versus number of ponds may suggest the overall Pearson residuals are increasing slightly as the number of ponds increases for some iterations (Figure 27), but these plots are difficult to interpret. As with typical residual diagnostics, patterns in these plots may suggest that some of the variability associated with that covariate is not being accounted for in the model. There does not

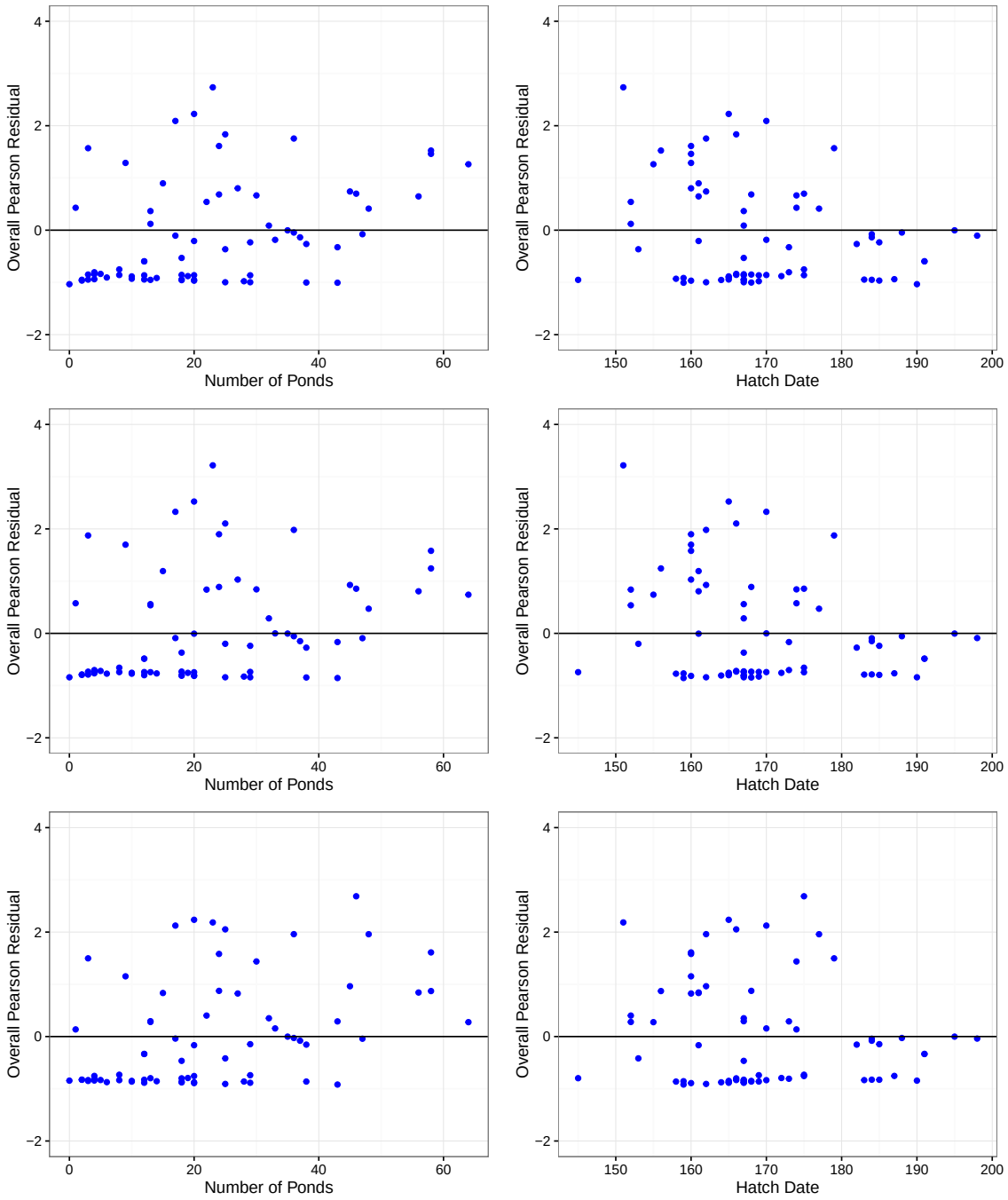

Figure 27: Overall Pearson residual plots from three random iterations (each row) of the initial zero-inflated model. The first column shows these versus number of ponds and the second column shows these versus hatch date.

appear to be any strong patterns in the overall Pearson residuals versus hatch date (Figure 27). It seems unusual that some residuals have a large, positive magnitude around 3 but that none of the residuals are less than -1. This could be due to the zero-inflation process of the model and the overall high probability of a zero count. Examining plots of the posterior predictive residuals is a way to assess whether this pattern is unusual or not. The posterior predictive residuals can also provide a reference to help in the general interpretation of these residual plots, but we do not explore these here.

The overall Pearson residual plots are potentially confusing because any missing covariate structure can be for the zero-inflated portion of the model or for the binomial process. This approach already includes a fairly complex structure to model the probability of duckling survival given the two covariates available. In other words, it seems reasonable that this model already accounts for the heterogeneity in the probability of ducklings survival (p) due to hatch date and number of ponds. This model, however, makes the assumption that the probability of a catastrophic event is constant for all broods. Using the hatch date and number of ponds variables to account for any heterogeneity in this probability may be more ecologically realistic and address any potential patterns in the overall Pearson residual plots.

To further explore residuals from this model we examine the binomial process and the zero-inflated process separately from one another. To assess whether the model structure for the probability of duckling survival in the binomial process is adequate, only broods that do not experience a catastrophic event are examined. However, occurrence of these events are a partially observed latent variable since for broods with zero observed ducklings we do not know whether this count is due to a catastrophic event or a zero from the binomial process, whereas we do know that broods with counts greater than zero did not experience a catastrophic event. Given the number of surviving ducklings for a brood, the conditional probability of

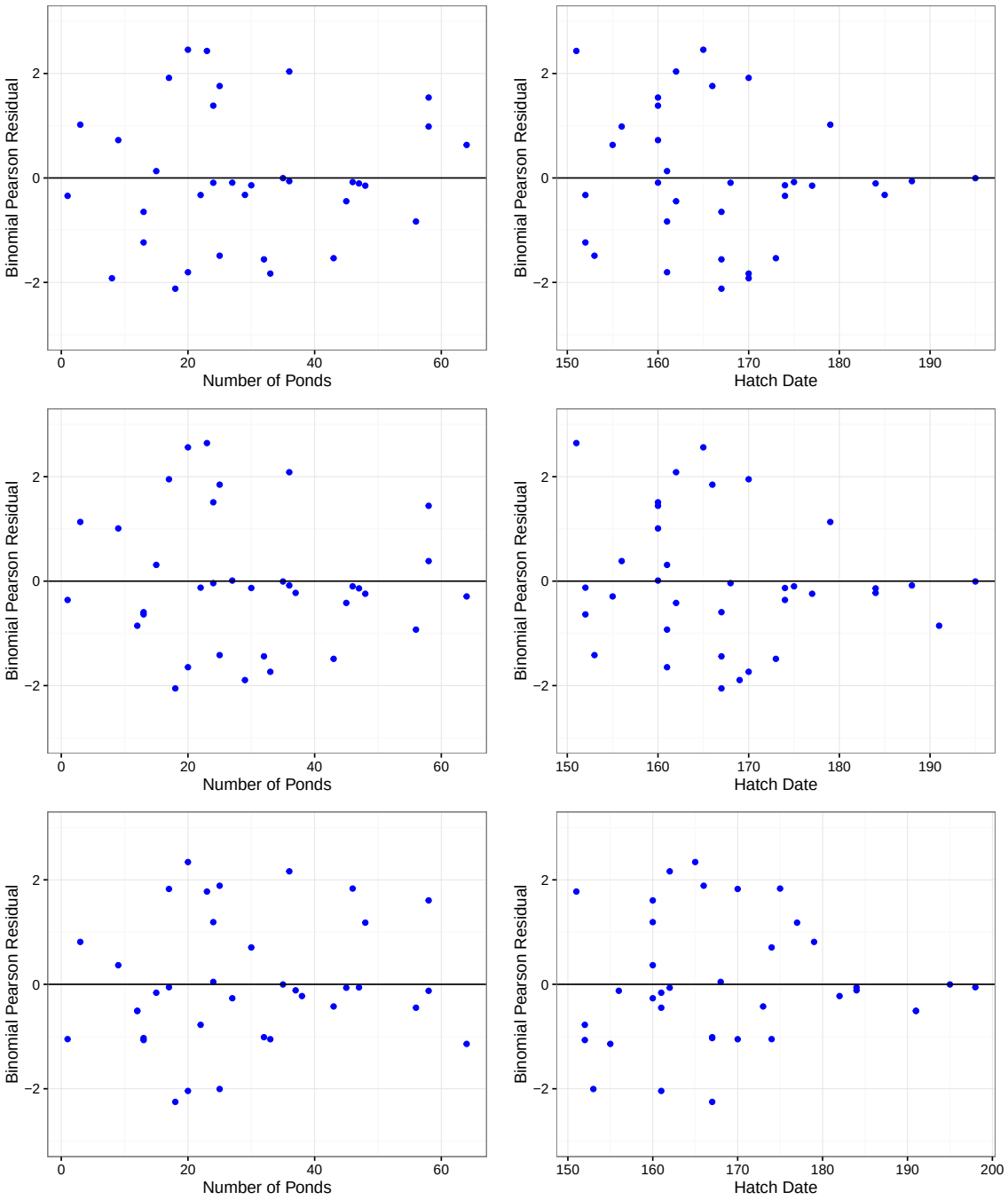

Figure 28: Binomial Pearson residuals from three random iterations (each row) of the initial zero-inflated model. No patterns are seen versus number of ponds or hatch date for these residuals.

a catastrophic event for that brood can be defined as

$$\Pr(Z_i = 1) = \frac{\pi(1-p_i)^{m_i}}{(1-\pi)+\pi(1-p_i)^{m_i}}$$

for broods with zero surviving ducklings and $\Pr(Z_i = 1) = 1$ for broods with more than zero ducklings. Here, Z_i is the partially latent variable with a 1 indicating no catastrophic event for brood i.

Using these probabilities, a Bernoulli random variable can be simulated for each brood at each iteration. Again, note that there is only uncertainty about this latent variable for broods with zero surviving ducklings. It should also be noted that, the following residual calculations are not true residuals since they are based on the partially latent variable indicating the occurrence of a catastrophic event and not a fully observable variable. Now, the binomial Pearson residuals can be calculated as

$$r_i^{bin} = \frac{Y_i - m_i p_i}{\sqrt{m_i p_i (1-p_i)}},$$

for broods with $Z_i = 1$ at a MCMC iteration. For these plots, we are not interested in the broods that experience a catastrophic event ($Z_i = 0$) since they do not provide information about the binomial process of the model. Examining the binomial Pearson residuals after conditioning on no catastrophic events provides a tool to specifically assess the model structure describing the probability of duckling survival. There are no patterns in the binomial Pearson residuals versus either hatch date or number of ponds for three random iterations (Figure 28), suggesting that the covariate structure for this part of the model is adequate.

To assess the model describing the probability of a catastrophic event, the Bernoulli Pearson residuals using the Z_i variables are examined for three random

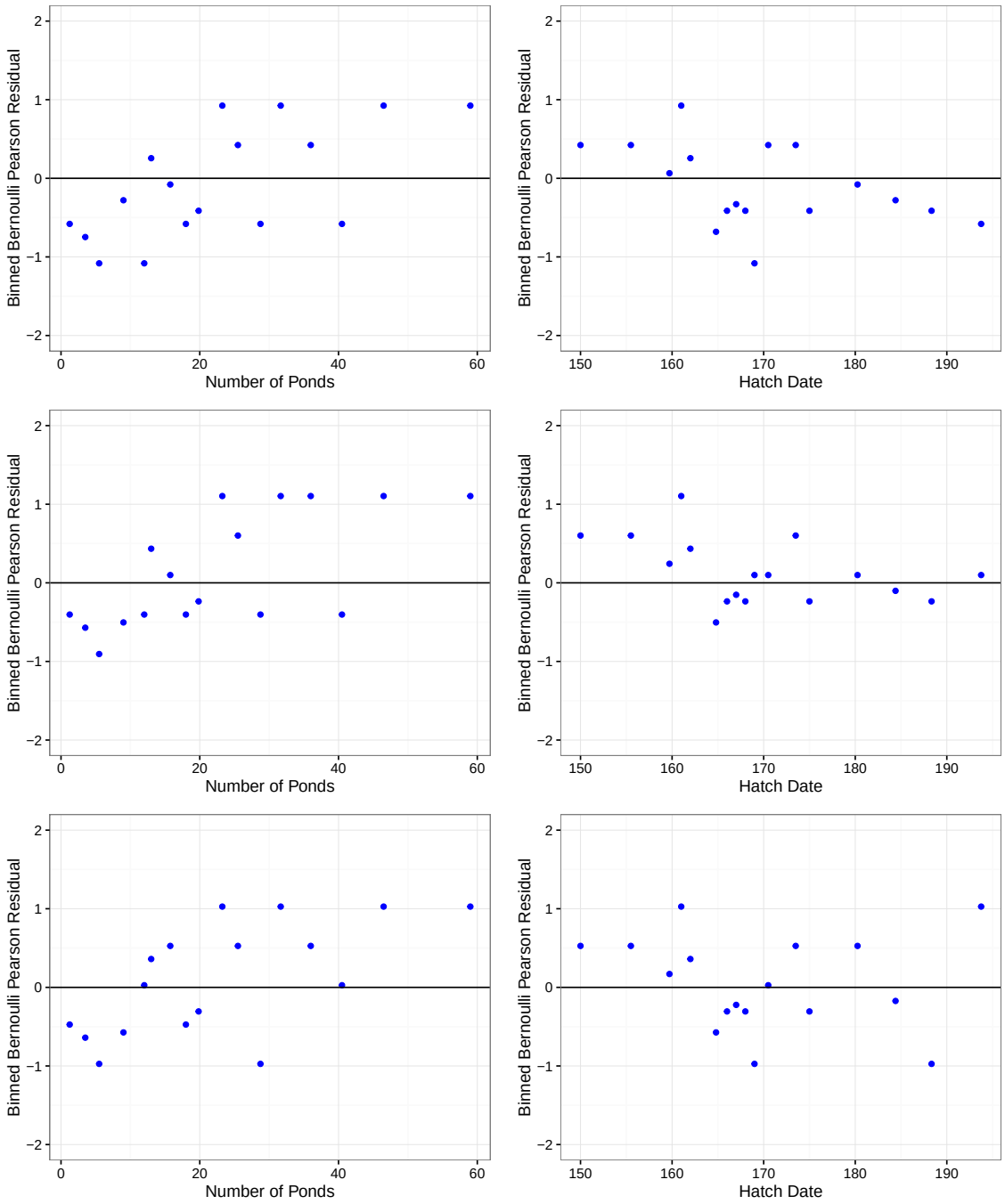

Figure 29: Binned Bernoulli Pearson residuals versus number of ponds and hatch date from three random iterations (each row) of the initial zero-inflated model. These residuals increase as number of ponds increases, but there is no strong pattern in these residuals versus hatch date.

iterations of the MCMC algorithm. These residuals are calculated as

$$r_i^{Bern} = \frac{Z_i - \pi}{\sqrt{\pi(1-\pi)}}$$

for every brood at each iteration. These raw Bernoulli residuals will not be informative because they are based on binary indicators, but can be interpreted more easily by averaging the residuals over specified bins (Gelman and Hill 2007). Here, we examine these residuals by first binning by number of ponds and then by hatch date. In each case, the average residual is plotted on the y-axis and the average value of the covariate on the x-axis, where the groups are made using quantiles of the covariates. The selection of the number of bins can change the interpretation of the plots, so it can be helpful to look at a variety of different bins when examining these plots. There appears to be a strong increasing pattern in the binned Bernoulli residuals versus number of ponds but no consistent patterns versus hatch date (Figure 29). The residuals from some iterations suggest a slight pattern versus hatch date, but this pattern is not present when additional iterations are examined (plots not shown). Overall, these additional residual plots indicate that including number of ponds in the logistic model for the probability of no catastrophic event may be needed for these data.

4.6 Zero-Inflated Binomial Model with Covariates for π

4.6.1 Model

The residual plots of the previous model (Section 4.5.4) suggest that the number of ponds variable may explain heterogeneity in the probability of a catastrophic event. This makes sense ecologically considering the likely reasons leading to catastrophic events for mallard broods. Hens may have difficulty feeding themselves in poor habitat conditions resulting in them being more stressed and more likely to abandon the

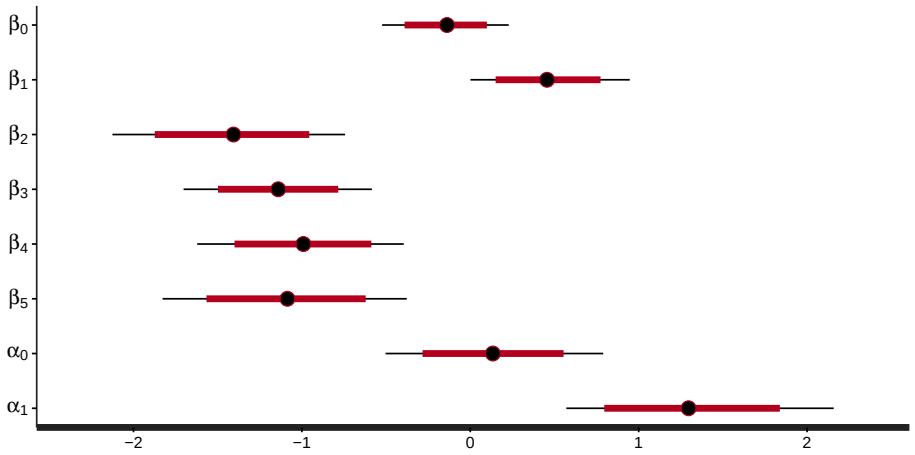

Figure 30: Summary of the posterior draws for each parameter from the zero-inflated model using number of ponds as an explanatory variable for the process generating zeroes. The posterior mean for each parameter is shown by the point (circle) along with the 80% PIs (thick, red) and 95% PIs (thin, black) depicted by lines.

brood. Additionally, mallards use ponds to avoid predators and a brood may have more success doing so when more ponds are near its nesting site. Using these ideas, the next model incorporates the number of ponds as an explanatory variable in the logit link for the probability of no catastrophic event, that is,

$$\text{logit}(\pi_i) = \alpha_0 + \alpha_1 NP_i,$$

with vague priors for these parameters - $\alpha_j \sim \text{N}(0, 900)$ for $j = 0, 1$. The remaining structure of the model is identical to that in Equations 4.8 to 4.12. The posterior distributions describing the probability of duckling survival are similar to those using the previous model (Section 4.5.1) and there is strong evidence that the coefficient for the number of ponds in the zero-inflated process differs from zero (Figure 30).

4.6.2 Results

The results of this model can be initially understood by looking at the zero-inflated and binomial processes individually. Using this model, the probability of

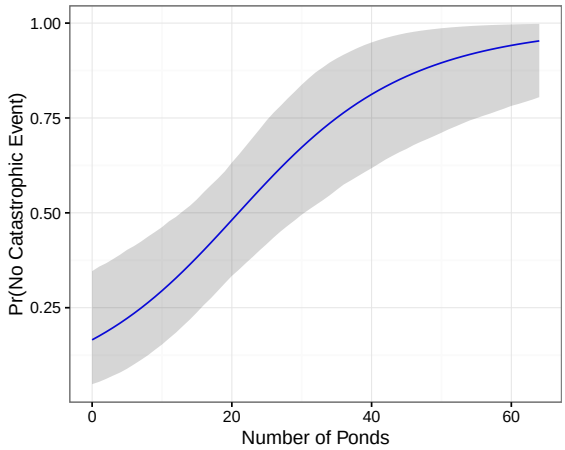

Figure 31: Based on the second zero-inflated model, the probability of a brood not being abandoned versus hatch date. The line shows the posterior mean for this relationship and the 95% PI is shaded.

no catastrophic event increases as the number of ponds increases (Figure 31). Recall that the covariates in this analysis are centered and standardized. Using the posterior mean, an increase of 15.6 ponds (one standard deviation) is associated with an increase in the odds of a brood not experiencing a catastrophic event by 3.67 times (95% PI

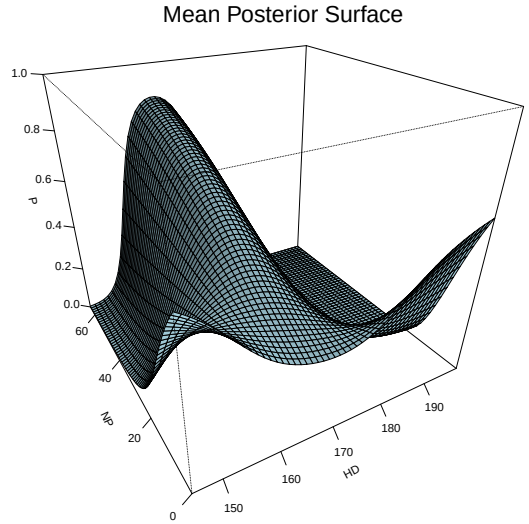

Figure 32: The mean posterior duckling survival probability surface for different combinations of hatch date and number of ponds based on the zero-inflated model with covariates for π. This plot is conditional on the brood not being abandoned.

from 1.77 to 8.67).

As with the previous zero-inflated model (Section 4.5.2), the parameters modeling duckling survival (βs) are conditional on a brood not experiencing a catastrophic event and are best interpreted with plots. The mean posterior surface of conditional duckling survival using this model appears similar in overall shape to those from the previous model fits (Figure 32). Again, the uncertainty in the conditional duckling survival probabilities is considerably less than the models including overdispersion using normal errors, but there is still substantial uncertainty in the probabilities for broods with a very few number of ponds and either a early or late hatch date (Figure 33). This is also consistent with the previous models and again illustrates some of the limitations of the information available in these data. We do not show the posterior predictive checks for this model here, but none of the assessments indicate a lack of agreement between the model and the observed data. This makes sense given the checks from the previous zero-inflated model (Section 4.5.3) do not indicate issues. Since this model is slightly more complex, it is unsurprising that it is also consistent with the observed data using the same summaries for posterior predictive checks.

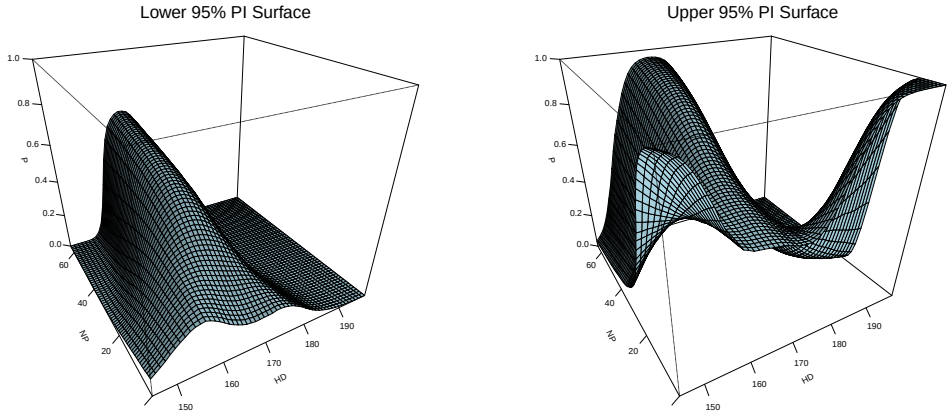

Figure 33: Based on the Zero-Inflated model, the 95% posterior interval duckling survival probability surface for different combinations of hatch date and number of ponds. Note that this plot is now conditional on the brood not being abandoned.

4.6.3 Residual Plots

The number of ponds variable is included in this model because some of the residual plots for the previous model (Section 4.5.4) indicate an increasing pattern with this variable. Specifically, the binned Bernoulli residuals suggest that including this covariate to model the probability of a catastrophic event is important for these data. We focus on the binned Bernoulli residuals for this model to assess whether this model appropriately accounts for the heterogeneity in the probability of a catastrophic event due to the number of ponds. For this model, there does not appear to be any patterns in these residuals versus either of the covariates (Figure 34).

Overall, these assessments indicate this model appears to produce posterior predictive datasets that are consistent with the observed data for these summaries. Additionally, there are no unusual patterns in the Pearson residual plots in comparison to the previous zero-inflated model. This suggests that there are no structural problems with this model and that these covariates adequately account for the heterogeneity in the probabilities of brood catastrophic events and conditional duckling survival. We will use the zero-inflated model with number of ponds as an explanatory variable for the probability of no catastrophic event for further inferences in the following section. While these plots help to understand the relationships between these covariates with conditional probability of duckling survival and with the probability of no catastrophic event, other quantities from this model may be of interest. The results of this model and these additional quantities are explored further in the following section.

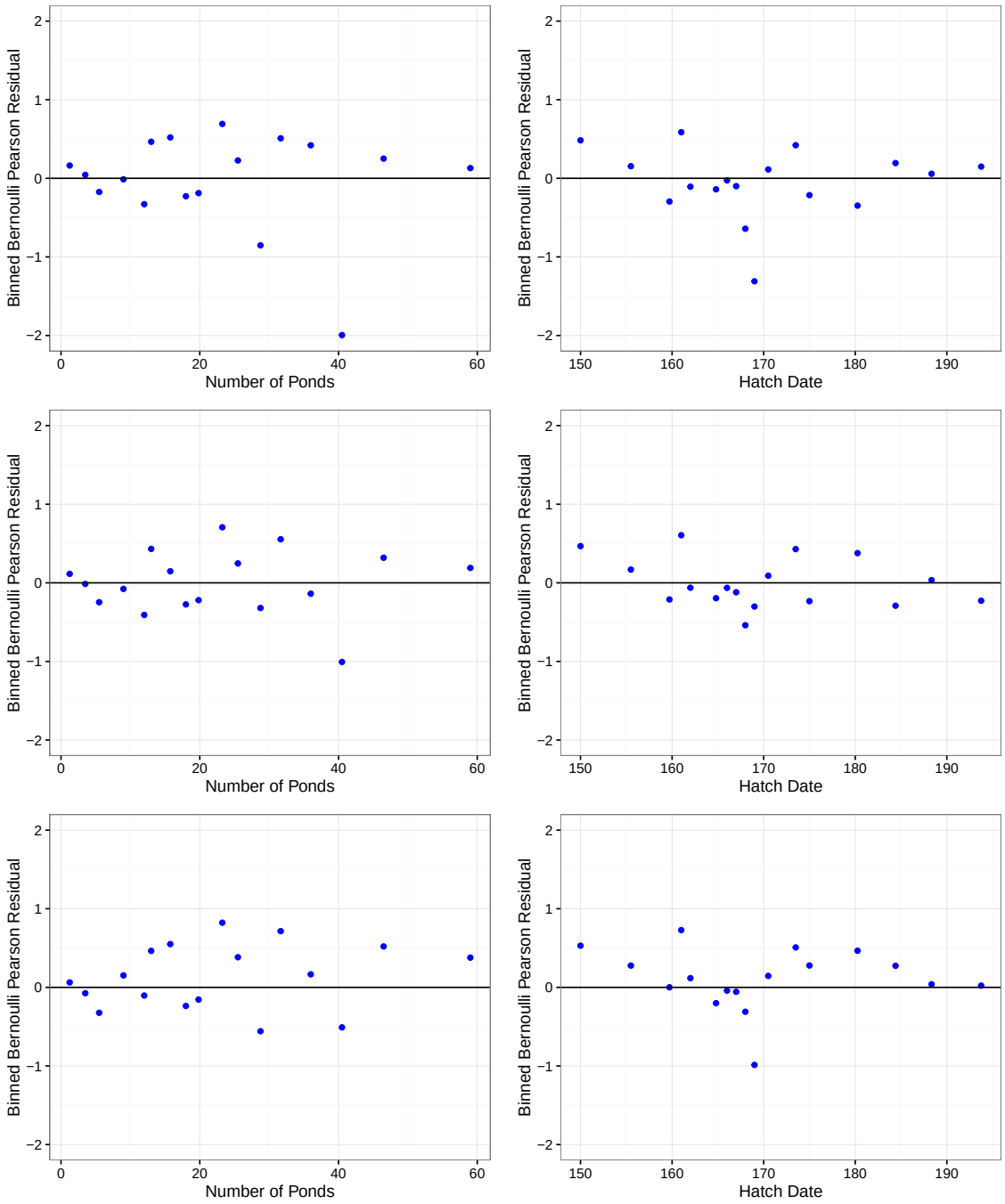

Figure 34: Binned Bernoulli Pearson residuals versus each covariate from three random iterations (each row) of the zero-inflated model with covariates for the probability of a catastrophic event. There is no evidence of strong patterns between the residuals and either covariate.

5 Discussion

5.1 Additional Inferences

Zero-inflated models explain the observed counts of surviving ducklings as a two stage process: a brood can experience a catastrophic event where all ducklings are killed at once, but if this does not happen, the survival of the ducklings for 30 days is driven by other factors. There appears to be evidence that the probability of no catastrophic event increases as the number of ponds increases. For the probability of duckling survival conditional on no catastrophic event, Equation 4.9 describes the covariate structure modeling this probability on the logit link. For this zero-inflated model (Section 4.6), explicitly examining the estimates for either of these processes may be of interest ecologically and useful for understanding how the number of ponds around a brood's nest impacts duckling survival. Additionally, we can understand and think about other parameters that may be of interest to researchers. In this section we will investigate the overall probability of duckling survival, the estimated optimal hatch date, and the probability of brood survival.

First, we examine the unconditional probability of duckling survival for different combinations of hatch date and number of ponds. In other words, we are interested in the probability of duckling survival accounting for both the probability that a catastrophic event occurs and the probability of duckling survival when no catastrophic event occurs. Mathematically, using the notation above (Equation 4.8) this is calculated by multiplying π_i with p_i where both of these quantities are modeled with covariates using the logit link. Using a Bayesian approach, the posterior distribution for this probability can be obtained for each combination of hatch date and number of ponds. This is done by simply multiplying these probabilities together for each saved iteration from the MCMC algorithm. Summarizing these posterior distributions provides inference about the probabilities of overall duckling survival

for different covariate patterns. Comparatively, describing the uncertainty for these probabilities using a ML approach is not as straightforward and requires the use of the Delta method to approximate confidence intervals.

The overall duckling survival probability surface we create using the posterior mean at each combination of hatch date and number of ponds is similar to those for previous models (Figure 35). Again, the probability of duckling survival increases as hatch date increases up to a point and then declines as hatch date increases further. Broods with very late hatch dates, after around 180, have an extremely low probability of survival regardless of the number of ponds around the nest site. The change in duckling survival probability as hatch date increases is less pronounced for broods with fewer ponds in comparison to broods with more ponds. Note that the benefit of more ponds is greater when examining overall duckling survival compared to the conditional probability of survival because we are now taking into account the probability of total brood loss from a catastrophic event. Since the probability of no catastrophic event increases as the number of ponds increases, a brood's surrounding

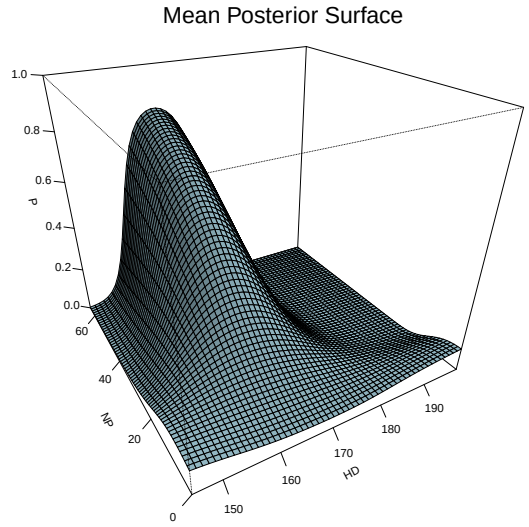

Figure 35: The posterior mean overall duckling survival probability surface for different combinations of hatch date and number of ponds based on the second zero-inflated model. These probabilities also account for catastrophic events.

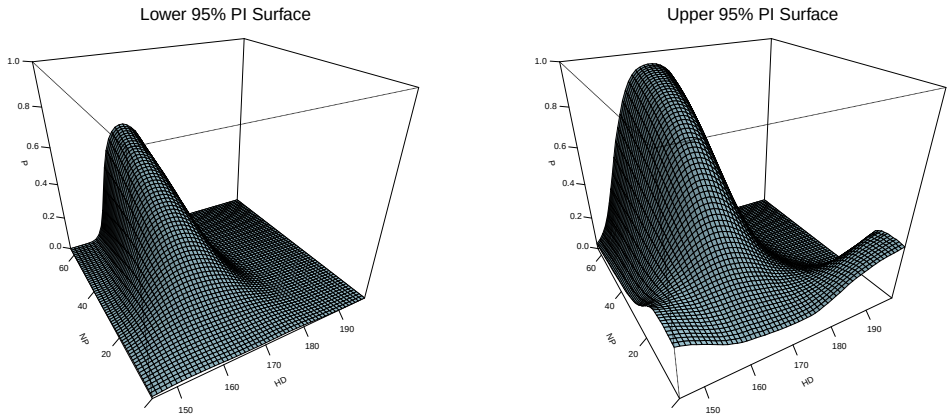

Figure 36: Based on the second zero-inflated model, the 95% posterior interval for the overall duckling survival probability surface for different combinations of hatch date and number of ponds. These probabilities include the probability of a catastrophic event.

wetland density has a larger impact in the overall probability of duckling survival than the conditional probability.

There is much less uncertainty for the surface describing the overall probability of duckling survival (Figure 36) than those from some of the previous models (Section 4.3.2). In particular, the folding of the corners of this surface plot for broods with few ponds and either early or late hatch dates is not as extreme. However, it appears that there is much more certainty that the overall probability of duckling survival is low compared to the PIs for the conditional probability of survival using the same model. In other words, we have more information about overall survival for broods with few ponds and a early or late hatch date, but this model does not distinguish which process in the model results in survival being so low.

Next, we examine the optimal hatch date for a given number of ponds using this model. This is also a quantity whose posterior distribution can be obtained using the posterior draws for the other model parameters. From the model structure for the probability of duckling survival on the logit scale (Equation 4.9), it can be shown

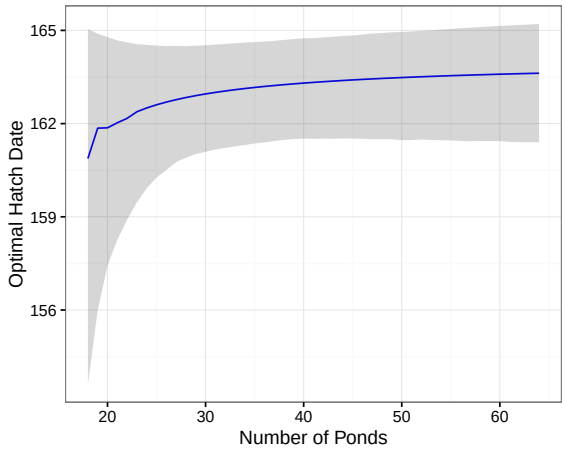

Figure 37: Posterior mean (line) and 95% PI (band) for the optimal hatch date for broods with 18-64 ponds near the nest site. Uncertainty for the optimal hatch date is extremely large for broods with fewer than 18 ponds.

that the optimal hatch date is

$$\text{HD}_{opt} = \frac{-(\beta_2 + \beta_4 \text{NP})}{2(\beta_3 + \beta_5 \text{NP})}$$

for a given number of ponds (NP). This calculation can be done for the range of the number of ponds variable in the observed data to calculate an optimal hatch date for every wetland density. However, for broods with less than 18 ponds, the uncertainty associated with the optimal quantity is extremely large. This makes sense using the results seen previously for this model - with fewer nearby ponds, duckling survival is estimated to be extremely low regardless of the hatch date. For broods with more than 18 ponds, the optimal hatch date is estimated to increase slightly as number of ponds increases using the posterior mean but overall a hatch date between 161 and 165 appears to be optimal once a brood has at least 30 nearby ponds (Figure 37).

The final quantity we examine is the probability of brood success defined as a brood having at least one surviving duckling after 30 days. This quantity is modeled in other analyses of duckling survival (see the following section). Here, we estimate this probability for a brood with 8 ducklings because that is the average number of eggs

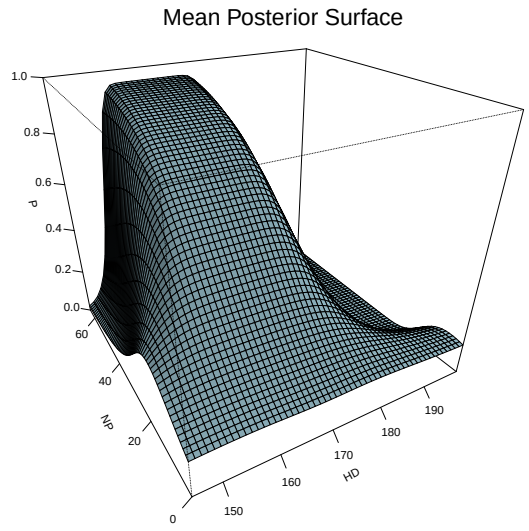

Figure 38: Mean posterior surface for the probability of brood success given 8 ducklings hatch for different combinations of hatch date and number of ponds.

for the broods in these data. Again, this quantity accounts for both the probability of a catastrophic event and the conditional probability of duckling survival. Brood survival is calculated as

$$\Pr(\text{Brood Survival}) = \pi_i(1 - (1 - p_i)^8),$$

where π_i and p_i vary depending on the explanatory variables. Overall, the probability of brood survival is quite high for a variety of covariate combinations, but quickly declines if the hatch date is either too early or too late (Figure 38). This pattern holds when accounting for the uncertainty associated with the probability of brood survival (Figure 39).

5.2 Comparison to other Duckling Survival Analyses

The inferences from Sections 4.6.2 and 5.1 can be compared to those from the original analysis of these data and other analyses of duckling survival. In the original analysis, Rotella and Ratti (1992) first use logistic regression to analyze a binary

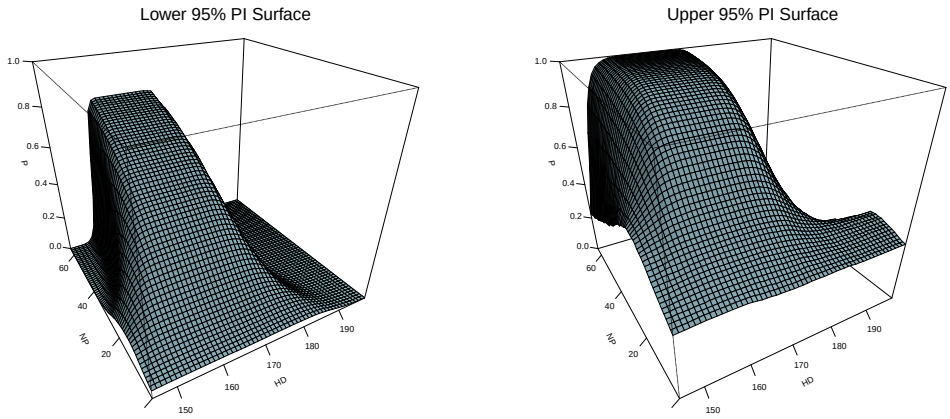

Figure 39: Upper and lower bounds of the 95% PI of the probability of brood success with 8 ducklings for different combinations of hatch date and number of ponds.

indicator for brood success and conclude that this is "directly related to wetland density and inversely related to hatching date" (pg. 499). Other analyses of duckling survival also focus on brood success (e.g., Talent, Jarvis, and Krapu 1983), but both ignore some of the available information because how many ducklings survive in each brood is not incorporated in the analysis. The zero-inflated model from Section 4.6 still allows inference to be made about brood success and how it is related to the covariates in this analysis. We are able to utilize all the information available in the duckling counts to inform the probability of a catastrophic event and the probability of duckling survival - each of which is related to brood success in the framework we present. A Bayesian approach to this analysis makes expressing uncertainty for derived parameters such as this much easier than it is under a ML approach.

The original analysis also compares the proportion of surviving ducklings across different categories of wetland density and hatch date using median tests (Rotella and Ratti 1992). Previous analyses of duckling counts utilize logistic regression but often account for overdispersion using a quasibinomial approach (Hoekman et al. 2004; Chouinard Jr, Arnold, and Haukos 2007; Amundson and Arnold 2011). In some of these analyses, the overdispersion parameter is estimated to be quite high and Hoekman et al. (2004) attribute this to "total brood loss over single intervals" of the

time periods examined. In this report, the analyses accounting for overdispersion with either the normal error terms or the beta-binomial regression result in unsatisfactory inferences because the resulting uncertainty for duckling survival is extremely large. The zero-inflated model, in comparison, presents a way to incorporate overdispersion by allowing for a process that directly leads to zeroes. In this way, the ecological process of catastrophic events resulting in total brood loss as described by others (Mauser, Jarvis, and Gilmer 1994; Pearse and Ratti 2004; Chouinard Jr, Arnold, and Haukos 2007) is directly accounted for in the model. Additionally, the posterior predictive checks of the zero-inflated model (Section 4.5.3) show no evidence of additional overdispersion in the duckling counts after incorporating the catastrophic events. Therefore, for this dataset, it appears that the overdispersion is a result of additional zeroes and not due to non-independence of ducklings within a brood as others speculate (Chouinard Jr, Arnold, and Haukos 2007). The zero-inflated model allows for overdispersion by modeling the zero counts from catastrophic events and provides a way to make ecologically meaningful inferences about that process as well as more precise inferences about duckling survival.

5.3 Conclusion

Explicitly accounting for catastrophic events with a zero-inflated binomial model is a more realistic representation of the ecological processes describing mallard duckling survival than assuming the counts are strictly binomially distributed. This is seen in the comparison of the posterior predictive checks for the various models we examine throughout this paper as well as descriptions of total brood loss by other researchers (Mauser, Jarvis, and Gilmer 1994; Pearse and Ratti 2004; Chouinard Jr, Arnold, and Haukos 2007). We see that modeling the catastrophic events which generate zero counts appears to address the overdispersion in the duckling counts and allows for more precise inference for how survival is related to hatch date and

number of ponds.

Bayesian inference is utilized throughout this report and highlights some of the advantages of this approach for analyzing these data. Many of the models we explore here can be fit using ML estimation but will likely require additional R packages or programming likelihood functions by hand to obtain estimates. For these data, inferences from the Bayesian and ML approaches are likely similar, as is the case for the first binomial logistic regression model. The primary advantages to a Bayesian approach for these data is seen in how the posterior distribution can be utilized to evaluate and understand a model more completely. Posterior predictive checks provide an intuitive and natural way to compare the observed data to a model. Here, they are a natural way to build models and address deficiencies that are seen when comparing the observed data to posterior predictive datasets. Posterior distributions from a Bayesian analysis also provide a way to make inference for many additional quantities of interest. In comparison, inference for these derived parameters under a ML approach relies on approximations and assumptions of asymptotic normality. For these reasons, the Bayesian analysis allows a more complete understanding of duckling survival and how it is related to the covariates of interest.

The zero-inflated model we use to analyze these data indicates that the number of ponds near a brood's nest site impacts duckling survival in two ways. First, a higher wetland density reduces the probability of a brood experiencing a catastrophic event where all of the ducklings are killed at once. This may be due to more food availability resulting in hens being less likely to abandon their broods as well as better habitat for predator avoidance. For broods that do not experience a catastrophic event, higher wetland density is also associated with an increase in duckling survival. This increase is more dramatic for broods within a window of moderate hatch dates. For broods outside these dates, there is not as much of an increase in duckling survival as wetland density increases. This suggests a timing aspect to the benefit of wetland

density where, for instance, food availability may be seasonal as well. Overall, using a zero-inflated model in the analysis of these data provides a more complete and ecologically realistic understanding of duckling survival.

References

Amundson, Courtney L and Todd W Arnold (2011). "The role of predator removal, density-dependence, and environmental factors on mallard duckling survival in North Dakota". *The Journal of Wildlife Management* 75.6, pp. 1330–1339.

Carpenter, Bob, Andrew Gelman, Matt Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Michael A. Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell (2016). "Stan: A probabilistic programming language". *Journal of Statistical Software*. in press.

Chouinard Jr, Michael P, Todd W Arnold, and DA Haukos (2007). "Survival and habitat use of mallard (Anas platyrhynchos) broods in the San Joaquin Valley, California". *The Auk* 124.4, pp. 1305–1316.

Dahl, David B. (2015). *xtable: Export Tables to LaTeX or HTML*. R package version 1.8-0. URL: `https://CRAN.R-project.org/package=xtable`.

Gelman, Andrew and Jennifer Hill (2007). *Data analysis using regression and multilevelhierarchical models*. Vol. 1. Cambridge University Press Cambridge.

Hoekman, Steven T, T Shane Gabor, Ron Maher, Henry R Murkin, and Llwellyn M Armstrong (2004). "Factors affecting survival of mallard ducklings in southern Ontario". *The Condor* 106.3, pp. 485–495.

Lele, Subhash R and Brian Dennis (2009). "Bayesian methods for hierarchical models: are ecologists making a Faustian bargain". *Ecological Applications* 19.3, pp. 581–584.

Mauser, David M, Robert L Jarvis, and David S Gilmer (1994). "Survival of radio-marked mallard ducklings in northeastern California". *The Journal of wildlife management*, pp. 82–87.

Pearse, Aaron T and John T Ratti (2004). "Effects of predator removal on mallard duckling survival". *Journal of Wildlife Management* 68.2, pp. 342–350.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/.

Ramsey, Fred and Daniel Schafer (2012). *The statistical sleuth: a course in methods of data analysis*. Cengage Learning.

Rotella, Jay J. and John T. Ratti (1992). "Mallard Brood Survival and Wetland Habitat Conditions in Southwestern Manitoba". *Journal of Wildlife Management* 53.3, pp. 499–507.

Stan Development Team (2016). *RStan: the R interface to Stan*. R package version 2.9.0. URL: http://mc-stan.org.

Talent, Larry G, Robert L Jarvis, and Gary L Krapu (1983). "Survival of mallard broods in south-central North Dakota". *Condor*, pp. 74–78.

Wickham, Hadley (2009). *ggplot2: elegant graphics for data analysis*. Springer New York. ISBN: 978-0-387-98140-6. URL: http://had.co.nz/ggplot2/book.

Wickham, Hadley and Romain Francois (2015). *dplyr: A Grammar of Data Manipulation*. R package version 0.4.3. URL: https://CRAN.R-project.org/package=dplyr.

A Appendix

A.1 Plots

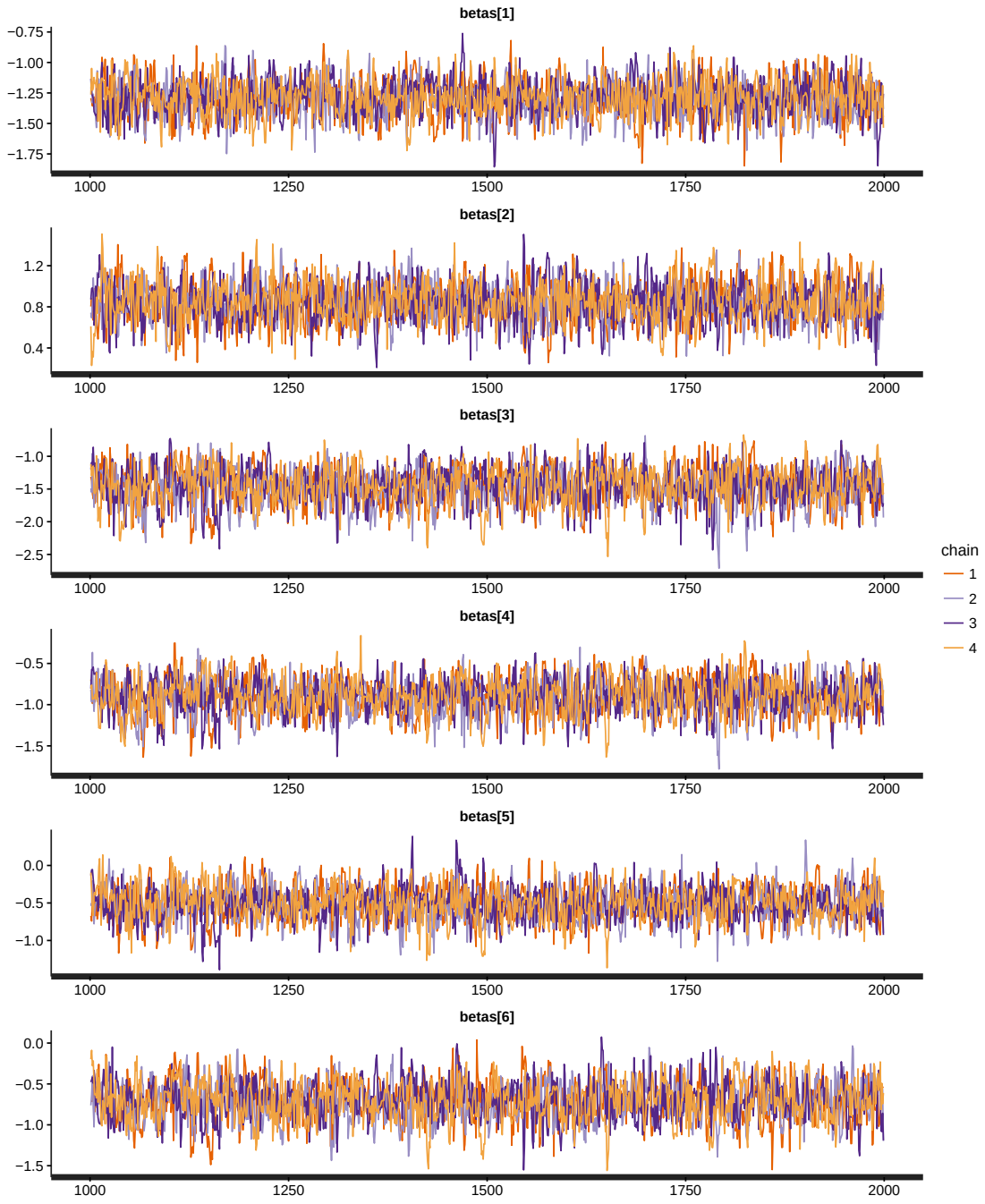

Figure 40: Traceplots for parameters in the Bayesian logistic regression model.

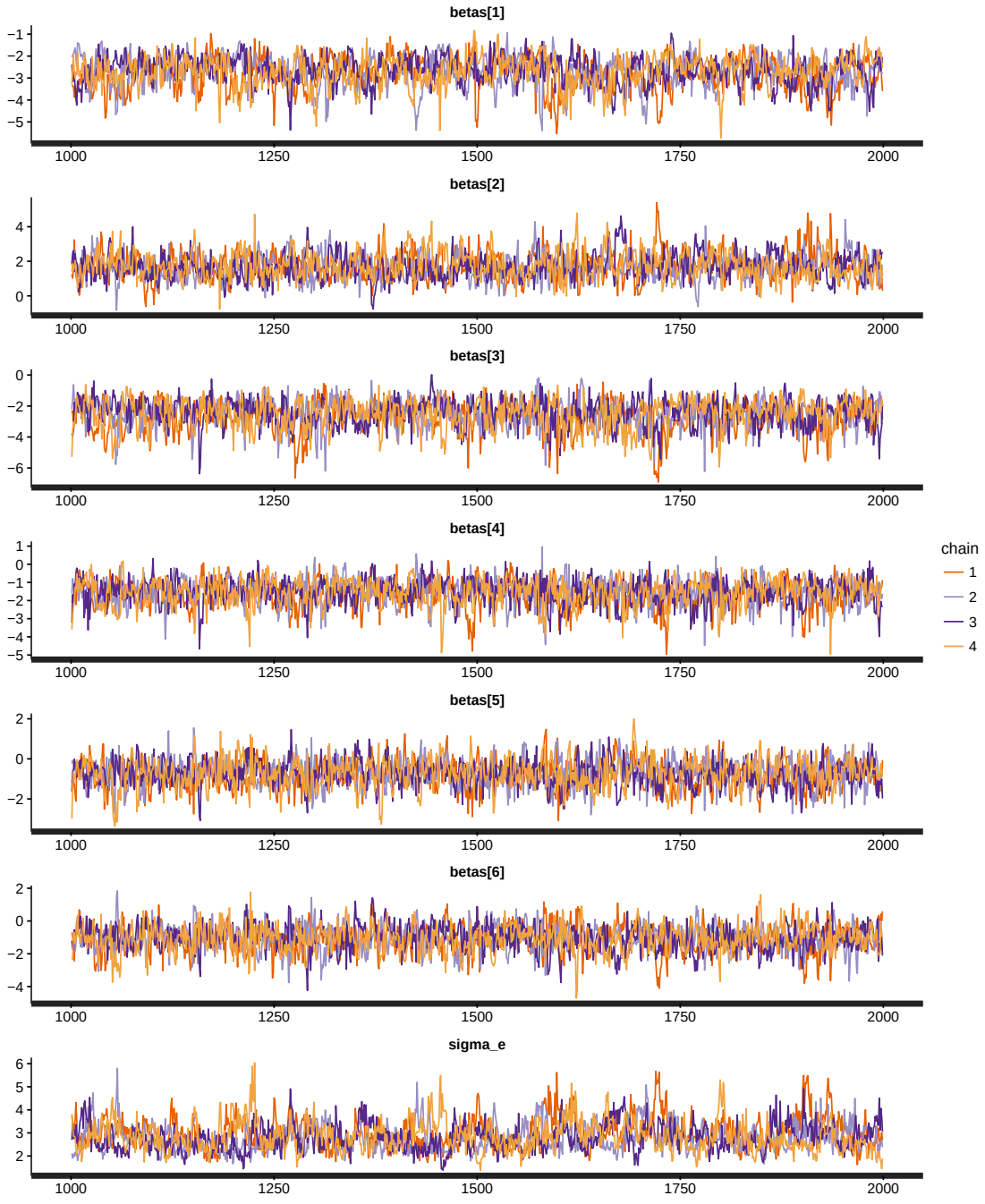

Figure 41: Traceplots for the parameters from the logistic regression with normal errors model.

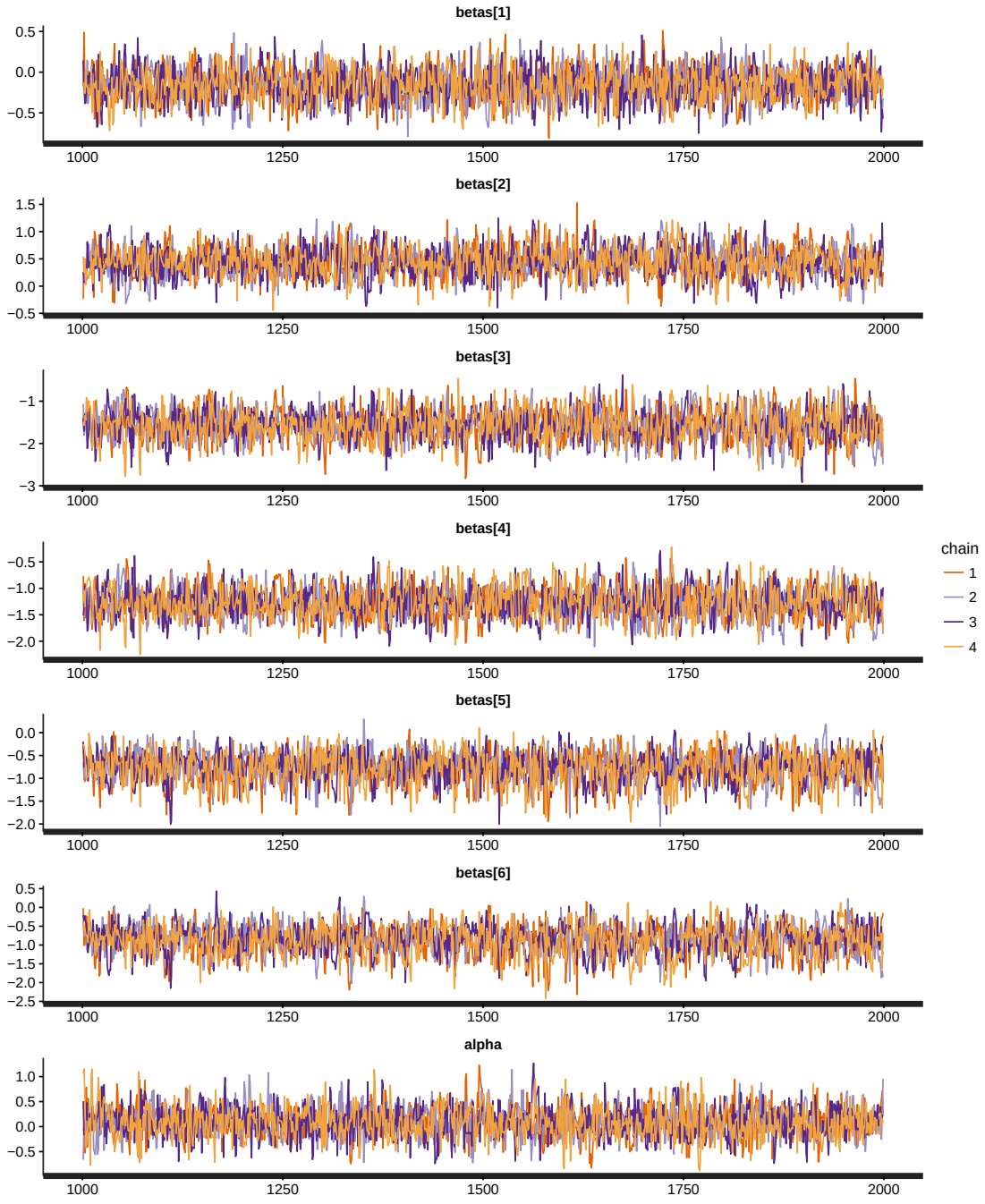

Figure 42: Traceplots for the regression coefficients from the first zero-inflated model.

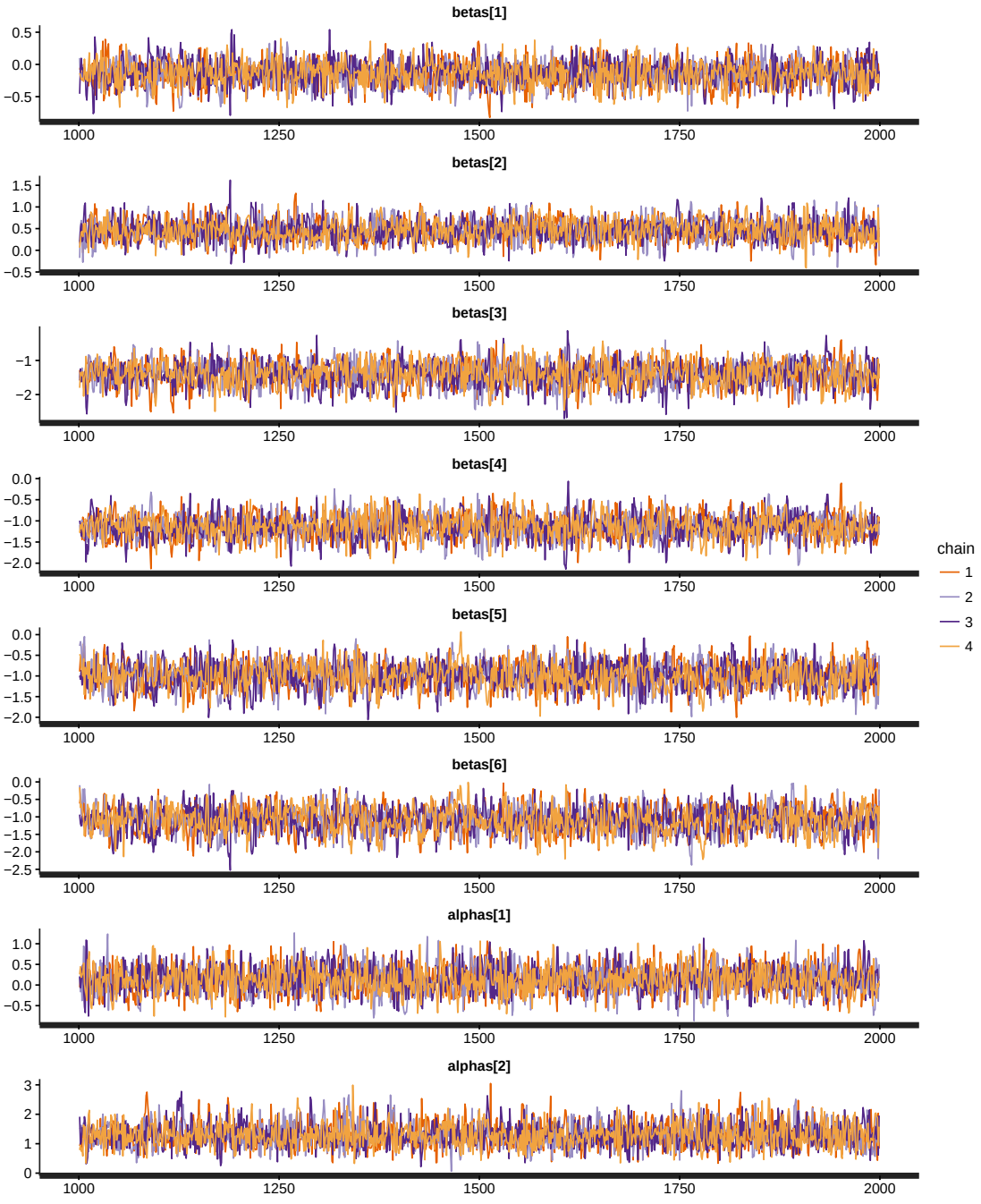

Figure 43: Traceplots for the regression coefficients from the zero-inflated model with the probability of a catastrophic event varying by number of ponds.

A.2 R Code

Exploratory Plots

```
opts_chunk$set(fig.width=5, fig.height=4, out.width='\\linewidth', dev='pdf',
               concordance=TRUE, size='footnotesize')
options(replace.assign=TRUE,width=112, digits = 3, max.print="72",
        show.signif.stars = FALSE)
setwd("C://Wilson/School/WritingProject/stan_models")
dd.full <- read.csv("DuckDataFull.csv")

library(dplyr)
library(rstan)
library(xtable)

dd.full <- dd.full %>% mutate(psurv=count/eggs, hdsq = hatchd^2,
                              yr.ind=ifelse(year>1988, "1989", "1987/1988") ) %>%
  mutate(emp.log = log( (count+0.5) / (eggs-count+0.5)))
```

```
ggplot(dd.full, aes(x=factor(year), y=nponds, fill=factor(year))) +
  geom_boxplot() + guides(fill=FALSE) +
  geom_point(position=position_jitter(width=0.1)) +
  xlab("Year") + ylab("Number of Ponds") + theme_bw() +
  scale_fill_manual(values=c("#e41a1c", "#377eb8", "#4daf4a"))
```

```
ggplot(dd.full, aes(x=nponds, y=emp.log, shape=factor(year), col=factor(year))) +
  geom_point(aes(fill=factor(year))) + facet_wrap(~yr.ind) +
  theme_bw() + xlab("Number of Ponds") +
  ylab("Empirical Logit") + guides(shape=FALSE, col=FALSE, fill=FALSE) +
  scale_shape_manual(values=c(21, 22, 23)) +
  scale_colour_manual(values=c("#e41a1c", "#377eb8", "#4daf4a")) +
  scale_fill_manual(values=c("#e41a1c", "#377eb8", "#4daf4a"))
```

```
ggplot(dd.full, aes(x=nponds, y=emp.log, shape=factor(year), col=factor(year))) +
  geom_point(aes(fill=factor(year))) +
  guides(shape=FALSE, col=FALSE, fill=FALSE) + xlab("Number of Ponds") +
  ylab("Empirical Logit") + theme_bw() +
  scale_shape_manual(values=c(21, 22, 23)) +
  scale_colour_manual(values=c("#e41a1c", "#377eb8", "#4daf4a")) +
  scale_fill_manual(values=c("#e41a1c", "#377eb8", "#4daf4a"))

ggplot(dd.full, aes(x=hatchd, y=emp.log, shape=factor(year), col=factor(year))) +
  geom_point(aes(fill=factor(year))) +
  guides(shape=FALSE, col=FALSE, fill=FALSE) + xlab("Hatch Date") +
  ylab("Empirical Logit") + theme_bw() +
  scale_shape_manual(values=c(21, 22, 23)) +
  scale_colour_manual(values=c("#e41a1c", "#377eb8", "#4daf4a")) +
  scale_fill_manual(values=c("#e41a1c", "#377eb8", "#4daf4a"))
```

```
ggplot(transform(dd.full, fct=cut(hatchd, c(0, 159, 168, 174, 200),
                                  labels=c("Hatch Date < 160",
                                           "Hatch Date 160-168",
                                           "Hatch Date 169-174",
                                           "Hatch Date > 174"))),
       aes(x=nponds, y=emp.log, col=factor(year), shape=factor(year))) +
  geom_point(aes(fill=factor(year))) + facet_wrap(~fct, nrow=2) +
  theme_bw() + xlab("Number of Ponds") + ylab("Empirical Logit") +
  guides(shape=FALSE, col=FALSE, fill=FALSE) +
  scale_shape_manual(values=c(21, 22, 23)) +
  scale_colour_manual(values=c("#e41a1c", "#377eb8", "#4daf4a")) +
  scale_fill_manual(values=c("#e41a1c", "#377eb8", "#4daf4a"))
```

```
ggplot(transform(dd.full, fct=cut(nponds, c(-1, 12, 20, 32, 65),
                                  labels=c("Ponds < 13",
                                           "Ponds 13-19",
                                           "Ponds 20-31",
                                           "Ponds > 31"))),
       aes(x=hatchd, y=emp.log, col=factor(year), shape=factor(year))) +
  geom_point(aes(fill=factor(year))) + facet_wrap(~fct, nrow=2) +
  theme_bw() + xlab("Hatch Date") + ylab("Empirical Logit") +
  guides(shape=FALSE, col=FALSE, fill=FALSE) +
  scale_shape_manual(values=c(21, 22, 23)) +
  scale_colour_manual(values=c("#e41a1c", "#377eb8", "#4daf4a")) +
  scale_fill_manual(values=c("#e41a1c", "#377eb8", "#4daf4a"))
```

Maximum Likelihood Approach

```
dd.full2 <- dd.full %>% dplyr::select(nponds, hatchd) %>% scale()
dd.full2 <- data.frame(eggs=dd.full$eggs, count=dd.full$count,
                       year=dd.full$year, dd.full2) %>%
  mutate(hdsq=hatchd^2)

glm2 <- glm(cbind(count, eggs-count) ~ nponds + hatchd + hdsq + nponds*hatchd +
              nponds*hdsq, data=dd.full2, family=binomial)

print(xtable(summary(glm2),
             caption='Coefficient summary of ML estimates
             using the glm function in R'),
      caption.placement=getOption("xtable.caption.placement", "top"))


plot(predict(glm2), rstandard(glm2))
plot(abs(rstandard(glm2)))
qqnorm(rstandard(glm2))
qqline(rstandard(glm2))
plot(dd.full$hatchd, rstandard(glm2))
plot(dd.full$nponds, rstandard(glm2))
plot(dd.full$totarea, rstandard(glm2))
```

```
plot(log(dd.full$totarea), rstandard(glm2))
plot(dd.full$perim, rstandard(glm2))
```

Bayesian Logistic Regression 4.2.1

```
stan1 <- stan_model("model.stan", model_name="initial")
stan.dat <- list(N=nrow(dd.full), count=dd.full$count, m=dd.full$eggs,
                 ponds=as.vector(scale(dd.full$nponds)),
                 hatchd=as.vector(scale(dd.full$hatchd)))
samps1 <- sampling(stan1, data=stan.dat)
```

```
mle.frame <- data.frame(mle=coefficients(glm2),
                        low=coefficients(glm2)-1.96*sqrt(diag(vcov(glm2))),
                        up =coefficients(glm2)+1.96*sqrt(diag(vcov(glm2))),
                        y  =c(6, 5, 4, 3, 2, 1))
plot(samps1, pars=c("betas"), point_est="mean") +
  scale_y_continuous(labels=c(expression(beta[5]),
                              expression(beta[4]),
                              expression(beta[3]),
                              expression(beta[2]),
                              expression(beta[1]),
                              expression(beta[0])),
                     breaks=c(1:6)) +
  geom_point(data=mle.frame, aes(x=mle, y=y+0.25), shape=15, size=3) +
  geom_errorbarh(data=mle.frame,
                 aes(y=y+0.25, xmin=low, xmax=up, x=mle, height=0))
```

4.2.2

```
betas1 <- extract(samps1, pars=c("betas"))$betas
z.fun1 <- function(hd.c, ponds.c, beta){
  hd <- (hd.c - mean(dd.full$hatchd)) / sd(dd.full$hatchd)
  ponds <- (ponds.c - mean(dd.full$nponds)) / sd(dd.full$nponds)
  return(plogis(cbind(1, ponds, hd, hd^2, ponds*hd, ponds*hd^2) %*% beta))
}
ponds.c <- c(0:64)
hd.c <- c(145:198)
z.mat1 <- apply(betas1, 1, function(x){outer(hd.c, ponds.c, z.fun1, beta=x)})
z.means <- apply(z.mat1, 1, mean)
z.means <- matrix(z.means, nrow=54, ncol=65)

#Surface
par(mai=c(0.25, 0, 0.25, 0))
persp(hd.c, ponds.c, z.means, theta=-30, phi=30, xlab="HD", ylab="NP", zlab="P",
      ltheta=120, shade=0.25, col="lightblue", lwd=0.25,
      main="", cex.lab=0.5, xaxs="i", zlim=c(0,1), box=T, axes=T,
      ticktype="detailed", cex.axis=0.5)
mtext("Mean Posterior Surface")
```

```
z.lo <- apply(z.mat1, 1, quantile, probs=0.025)
z.lo <- matrix(z.lo, nrow=54, ncol=65)
z.up <- apply(z.mat1, 1, quantile, probs=0.975)
z.up <- matrix(z.up, nrow=54, ncol=65)

#lower
par(mai=c(0.25, 0.1, 0.25, 0.1))
persp(hd.c, ponds.c, z.lo, theta=-30, phi=30, xlab="HD", ylab="NP", zlab="P",
      ltheta=120, shade=0.25, col="lightblue", lwd=0.25,
      main="", cex.lab=0.5, xaxs="i", zlim=c(0,1),
      ticktype="detailed", cex.axis=0.5)
mtext("Lower 95% PI Surface")

#upper
persp(hd.c, ponds.c, z.up, theta=-30, phi=30, xlab="HD", ylab="NP", zlab="P",
      ltheta=120, shade=0.25, col="lightblue", lwd=0.25,
      main="", cex.lab=0.5, xaxs="i", zlim=c(0,1),
      ticktype="detailed", cex.axis=0.5)
mtext("Upper 95% PI Surface")
```

4.2.3

```
ponds <- dd.full$nponds
hatchd <- dd.full$hatchd
n <- dd.full$eggs
y <- dd.full$count

p.class <- ifelse(ponds<25, 1, 2)
d.class <- ifelse(hatchd<160, 1, ifelse(hatchd>170, 3, 2))

y.sim <- extract(samps1, pars=c("y_rep"))$y_rep

m.p1d1 <- m.p2d1 <- m.p1d2 <- m.p2d2 <- m.p1d3 <- m.p2d3 <- NULL
for(i in 1:4000){
  m.p1d1[i] <- mean(y.sim[i,which(p.class==1&d.class==1)] /
                    n[which(p.class==1&d.class==1)])
  m.p2d1[i] <- mean(y.sim[i,which(p.class==2&d.class==1)] /
                    n[which(p.class==2&d.class==1)])
  m.p1d2[i] <- mean(y.sim[i,which(p.class==1&d.class==2)] /
                    n[which(p.class==1&d.class==2)])
  m.p2d2[i] <- mean(y.sim[i,which(p.class==2&d.class==2)] /
                    n[which(p.class==2&d.class==2)])
  m.p1d3[i] <- mean(y.sim[i,which(p.class==1&d.class==3)] /
                    n[which(p.class==1&d.class==3)])
  m.p2d3[i] <- mean(y.sim[i,which(p.class==2&d.class==3)] /
                    n[which(p.class==2&d.class==3)])
}

post.mean.frame <- data.frame(means=c(m.p1d1, m.p2d1,
                                      m.p1d2, m.p2d2,
                                      m.p1d3, m.p2d3),
```

```
                              p.class=rep(rep(c("Number of Ponds 0-24",
                                                "Number of Ponds 25-64"),
                                              each=4000), 3),
                              d.class=rep(c("Hatch Date 145-160",
                                            "Hatch Date 160-170",
                                            "Hatch Date 170-198"), each=8000))
vline.frame <- data.frame(
  x=c(mean(y[which(p.class==1&d.class==1)]/n[which(p.class==1&d.class==1)]),
      mean(y[which(p.class==2&d.class==1)]/n[which(p.class==2&d.class==1)]),
      mean(y[which(p.class==1&d.class==2)]/n[which(p.class==1&d.class==2)]),
      mean(y[which(p.class==2&d.class==2)]/n[which(p.class==2&d.class==2)]),
      mean(y[which(p.class==1&d.class==3)]/n[which(p.class==1&d.class==3)]),
      mean(y[which(p.class==2&d.class==3)]/n[which(p.class==2&d.class==3)])),
  p.class=rep(rep(c("Number of Ponds 0-24",
                    "Number of Ponds 25-64"), each=1), 3),
  d.class=rep(c("Hatch Date 145-160",
                "Hatch Date 160-170",
                "Hatch Date 170-198"), each=2))

ggplot(post.mean.frame, aes(x=means)) + facet_grid(p.class ~ d.class) +
  geom_histogram(aes(y=..density..),
                 binwidth=0.01, col="blue", alpha=0.4, size=0.1, fill="blue") +
  theme_bw() +
  xlab("Average Proportion") + ylab("Density") +
  geom_vline(aes(xintercept=x), vline.frame)
```

```
zero.sim <- NULL
for(i in 1:4000){
  zero.sim[i] <- sum(y.sim[i,]==0)
}
zero.counts <- as.vector(table(zero.sim))
zero.values <- dimnames(table(zero.sim))$zero.sim
zero.frame <- data.frame(y=zero.counts, x=as.numeric(zero.values))

all.sim <- NULL
for(i in 1:4000){
  all.sim[i] <- sum(y.sim[i,]==n)
}
all.counts <- as.vector(table(all.sim))
all.values <- dimnames(table(all.sim))$all.sim
all.frame <- data.frame(y=all.counts, x=as.numeric(all.values))

ggplot(zero.frame, aes(y=y, x=x)) + geom_point(col="blue") +
  geom_segment(aes(xend=x, yend=0), col="blue", lineend="round") +
  theme_bw() + xlab("Number of Broods") + ylab("Count") +
  ggtitle("Broods with Zero Surviving Ducklings") +
  theme(plot.title=element_text(face="plain")) +
  geom_vline(xintercept=sum(y==0), linetype="longdash")
ggplot(all.frame, aes(y=y, x=x)) + geom_point(col="blue") +
  geom_segment(aes(xend=x, yend=0), col="blue", lineend="round") +
  theme_bw() + xlab("Number of Broods") + ylab("Count") +
  ggtitle("Broods with All Ducklings Surviving") +
```

```
  geom_vline(xintercept=sum(y==n), linetype="longdash")
```

```
early0.sim <- late0.sim <- NULL
for(i in 1:4000){
  early0.sim[i] <- sum(y.sim[i, which(hatchd < 180)] == 0)
  late0.sim[i] <- sum(y.sim[i, which(hatchd > 179)] == 0)
}
early0.counts <- as.vector(table(early0.sim))
early0.values <- dimnames(table(early0.sim))$early0.sim
late0.counts <- as.vector(table(late0.sim))
late0.values <- dimnames(table(late0.sim))$late0.sim
split0.frame <- data.frame(y=c(early0.counts, late0.counts),
                           x=c(as.numeric(early0.values),
                               as.numeric(late0.values)),
                           time=c(rep("Hatch Date 145-179", length(early0.counts)),
                                  rep("Hatch Date 180-198", length(late0.counts))))

vline.split <- data.frame(x = c(sum(y[which(hatchd < 180)] == 0),
                                sum(y[which(hatchd > 179)] == 0)),
                          time=c("Hatch Date 145-179", "Hatch Date 180-198"))

ggplot(split0.frame, aes(y=y, x=x)) + geom_point(col="blue") +
  geom_segment(aes(xend=x, yend=0), col="blue", lineend="round") +
  theme_bw() + xlab("Number of Broods") + ylab("Count") +
  facet_grid(~time, scales="free") +
  theme(plot.title=element_text(face="plain")) +
  geom_vline(aes(xintercept=x), linetype="longdash", data=vline.split)
```

Accounting for Overdispersion using Normal Errors 4.3.1

```
stan2 <- stan_model("model2.stan", model_name="overdispersed")
samps2 <- sampling(stan2, data=stan.dat)
```

```
plot(samps2, pars=c("betas", "sigma_e"), point_est="mean") +
  scale_y_continuous(labels=c(expression(sigma),
                              expression(beta[5]),
                              expression(beta[4]),
                              expression(beta[3]),
                              expression(beta[2]),
                              expression(beta[1]),
                              expression(beta[0])),
                     breaks=c(1:7))
```

4.3.2

```
betas2 <- extract(samps2, pars=c("betas"))$betas
z.mat2 <- apply(betas2, 1, function(x){outer(hd.c, ponds.c, z.fun1, beta=x)})
z.means2 <- apply(z.mat2, 1, mean)
z.means2 <- matrix(z.means2, nrow=54, ncol=65)

#Surface
par(mai=c(0.25, 0, 0.25, 0))
persp(hd.c, ponds.c, z.means2, theta=-30, phi=30, xlab="HD", ylab="NP", zlab="P",
      ltheta=120, shade=0.25, col="lightblue", lwd=0.25,
      main="", cex.lab=0.5, xaxs="i", zlim=c(0,1),
      ticktype="detailed", cex.axis=0.5)
mtext("Posterior Mean Surface")
```

```
z.lo2 <- apply(z.mat2, 1, quantile, probs=0.025)
z.lo2 <- matrix(z.lo2, nrow=54, ncol=65)
z.up2 <- apply(z.mat2, 1, quantile, probs=0.975)
z.up2 <- matrix(z.up2, nrow=54, ncol=65)

#lower
par(mai=c(0, 0.1, 0.25, 0.1))
persp(hd.c, ponds.c, z.lo2, theta=-30, phi=30, xlab="HD", ylab="NP", zlab="P",
      ltheta=120, shade=0.25, col="lightblue", lwd=0.25,
      main="", cex.lab=0.5, xaxs="i", zlim=c(0,1),
      ticktype="detailed", cex.axis=0.5)
mtext("Lower 95% PI Surface")

#upper
persp(hd.c, ponds.c, z.up2, theta=-30, phi=30, xlab="HD", ylab="NP", zlab="P",
      ltheta=120, shade=0.25, col="lightblue", lwd=0.25,
      main="", cex.lab=0.5, xaxs="i", zlim=c(0,1),
      ticktype="detailed", cex.axis=0.5)
mtext("Upper 95% PI Surface")
```

4.3.3

```
y.sim2 <- extract(samps2, pars=c("y_rep"))$y_rep

m2.p1d1 <- m2.p2d1 <- m2.p1d2 <- m2.p2d2 <- m2.p1d3 <- m2.p2d3 <- NULL
for(i in 1:4000){
  m2.p1d1[i] <- mean(y.sim2[i,which(p.class==1&d.class==1)] /
                       n[which(p.class==1&d.class==1)])
  m2.p2d1[i] <- mean(y.sim2[i,which(p.class==2&d.class==1)] /
                       n[which(p.class==2&d.class==1)])
  m2.p1d2[i] <- mean(y.sim2[i,which(p.class==1&d.class==2)] /
                       n[which(p.class==1&d.class==2)])
  m2.p2d2[i] <- mean(y.sim2[i,which(p.class==2&d.class==2)] /
                       n[which(p.class==2&d.class==2)])
  m2.p1d3[i] <- mean(y.sim2[i,which(p.class==1&d.class==3)] /
                       n[which(p.class==1&d.class==3)])
```

```
  m2.p2d3[i] <- mean(y.sim2[i,which(p.class==2&d.class==3)] /
                       n[which(p.class==2&d.class==3)])
}

post.mean.frame2 <- data.frame(means=c(m2.p1d1, m2.p2d1,
                                       m2.p1d2, m2.p2d2,
                                       m2.p1d3, m2.p2d3),
                               p.class=rep(rep(c("Number of Ponds 0-24",
                                                 "Number of Ponds 25-64"),
                                               each=4000), 3),
                               d.class=rep(c("Hatch Date 145-160",
                                             "Hatch Date 160-170",
                                             "Hatch Date 170-198"), each=8000))


ggplot(post.mean.frame2, aes(x=means)) + facet_grid(p.class ~ d.class) +
  geom_histogram(aes(y=..density..),
                 binwidth=0.01, col="blue", alpha=0.4, size=0.1, fill="blue") +
  theme_bw() +
  xlab("Average Proportion") + ylab("Density") +
  geom_vline(aes(xintercept=x), vline.frame)
```

```
zero.sim2 <- NULL
for(i in 1:4000){
  zero.sim2[i] <- sum(y.sim2[i,]==0)
}
zero.counts2 <- as.vector(table(zero.sim2))
zero.values2 <- dimnames(table(zero.sim2))$zero.sim2
zero.frame2 <- data.frame(y=zero.counts2, x=as.numeric(zero.values2))

all.sim2 <- NULL
for(i in 1:4000){
  all.sim2[i] <- sum(y.sim2[i,]==n)
}
all.counts2 <- as.vector(table(all.sim2))
all.values2 <- dimnames(table(all.sim2))$all.sim2
all.frame2 <- data.frame(y=all.counts2, x=as.numeric(all.values2))

ggplot(zero.frame2, aes(y=y, x=x)) + geom_point(col="blue") +
  geom_segment(aes(xend=x, yend=0), col="blue", lineend="round") +
  theme_bw() + xlab("Number of Broods") + ylab("Count") +
  ggtitle("Broods with Zero Surviving Ducklings") +
  theme(plot.title=element_text(face="plain")) +
  geom_vline(xintercept=sum(y==0), linetype="longdash")
ggplot(all.frame2, aes(y=y, x=x)) + geom_point(col="blue") +
  geom_segment(aes(xend=x, yend=0), col="blue", lineend="round") +
  theme_bw() + xlab("Number of Broods") + ylab("Count") +
  ggtitle("Broods with All Ducklings Surviving") +
  geom_vline(xintercept=sum(y==n), linetype="longdash")
```

```
early0.sim2 <- late0.sim2 <- NULL
for(i in 1:4000){
  early0.sim2[i] <- sum(y.sim2[i, which(hatchd < 180)] == 0)
  late0.sim2[i] <- sum(y.sim2[i, which(hatchd > 179)] == 0)
}
early0.counts2 <- as.vector(table(early0.sim2))
early0.values2 <- dimnames(table(early0.sim2))$early0.sim2
late0.counts2 <- as.vector(table(late0.sim2))
late0.values2 <- dimnames(table(late0.sim2))$late0.sim2
split0.frame2 <- data.frame(y=c(early0.counts2, late0.counts2),
                            x=c(as.numeric(early0.values2),
                                as.numeric(late0.values2)),
                            time=c(rep("Hatch Date 145-179",
                                       length(early0.counts2)),
                                   rep("Hatch Date 180-198",
                                       length(late0.counts2))))

ggplot(split0.frame2, aes(y=y, x=x)) + geom_point(col="blue") +
  geom_segment(aes(xend=x, yend=0), col="blue", lineend="round") +
  theme_bw() + xlab("Number of Broods") + ylab("Count") +
  facet_grid(~time, scales="free") +
  theme(plot.title=element_text(face="plain")) +
  geom_vline(aes(xintercept=x), linetype="longdash", data=vline.split)
```

Zero-Inflated Binomial Approach 4.5.1

```
stan3 <- stan_model("zero_infl.stan",
                    model_name="zero_inflated")
samps3 <- sampling(stan3, data=stan.dat)
```

```
table3 <- summary(samps3, pars=c("betas", "alpha"))$summary[, -c(9, 10)]
dimnames(table3)[[1]] <- c("$\\beta_0$", "$\\beta_1$(NP)", "$\\beta_2$(HD)",
                           "$\\beta_3$(HD$^2$)", "$\\beta_4$(NP$\\cdot$HD)",
                           "$\\beta_5$(NP$\\cdot$HD$^2$)", "$\\alpha$")
dimnames(table3)[[2]] <- c("Mean", "SE mean", "SD",
                           "2.5\\%", "25\\%", "50\\%", "75\\%", "97.5\\%")
print(xtable(table3, digits=2,
             caption='Summary of the posterior draws for each parameter
             from the Zero-Inflated Bayesian model'),
      caption.placement=getOption("xtable.caption.placement", "top"),
      sanitize.text.function=function(x) x)
```

```
plot(samps3, pars=c("betas", "alpha"), point_est="mean") +
  scale_y_continuous(labels=c(expression(alpha),
                              expression(beta[5]),
                              expression(beta[4]),
                              expression(beta[3]),
```

```
                              expression(beta[2]),
                              expression(beta[1]),
                              expression(beta[0])),
                      breaks=c(1:7))
```

4.5.2

```
betas3 <- extract(samps3, pars=c("betas"))$betas
z.mat3 <- apply(betas3, 1, function(x){outer(hd.c, ponds.c, z.fun1, beta=x)})
z.means3 <- apply(z.mat3, 1, mean)
z.means3 <- matrix(z.means3, nrow=54, ncol=65)

#Surface
par(mai=c(0.25, 0, 0.25, 0))
persp(hd.c, ponds.c, z.means3, theta=-30, phi=30, xlab="HD", ylab="NP", zlab="P",
      ltheta=120, shade=0.25, col="lightblue", lwd=0.25,
      main="", cex.lab=0.5, xaxs="i", zlim=c(0,1),
      ticktype="detailed", cex.axis=0.5)
mtext("Mean Posterior Surface")
```

```
z.lo3 <- apply(z.mat3, 1, quantile, probs=0.025)
z.lo3 <- matrix(z.lo3, nrow=54, ncol=65)
z.up3 <- apply(z.mat3, 1, quantile, probs=0.975)
z.up3 <- matrix(z.up3, nrow=54, ncol=65)

#lower
par(mai=c(0, 0.1, 0.25, 0.1))
persp(hd.c, ponds.c, z.lo3, theta=-30, phi=30, xlab="HD", ylab="NP", zlab="P",
      ltheta=120, shade=0.25, col="lightblue", lwd=0.25,
      main="", cex.lab=0.5, xaxs="i", zlim=c(0,1),
      ticktype="detailed", cex.axis=0.5)
mtext("Lower 95% PI Surface")
#upper
persp(hd.c, ponds.c, z.up3, theta=-30, phi=30, xlab="HD", ylab="NP", zlab="P",
      ltheta=120, shade=0.25, col="lightblue", lwd=0.25,
      main="", cex.lab=0.5, xaxs="i", zlim=c(0,1),
      ticktype="detailed", cex.axis=0.5)
mtext("Upper 95% PI Surface")
```

4.5.3

```
y.sim3 <- extract(samps3, pars=c("y_rep2"))$y_rep2

m3.p1d1 <- m3.p2d1 <- m3.p1d2 <- m3.p2d2 <- m3.p1d3 <- m3.p2d3 <- NULL
for(i in 1:4000){
  m3.p1d1[i] <- mean(y.sim3[i,which(p.class==1&d.class==1)] /
                       n[which(p.class==1&d.class==1)])
```

```
  m3.p2d1[i] <- mean(y.sim3[i,which(p.class==2&d.class==1)] /
                       n[which(p.class==2&d.class==1)])
  m3.p1d2[i] <- mean(y.sim3[i,which(p.class==1&d.class==2)] /
                       n[which(p.class==1&d.class==2)])
  m3.p2d2[i] <- mean(y.sim3[i,which(p.class==2&d.class==2)] /
                       n[which(p.class==2&d.class==2)])
  m3.p1d3[i] <- mean(y.sim3[i,which(p.class==1&d.class==3)] /
                       n[which(p.class==1&d.class==3)])
  m3.p2d3[i] <- mean(y.sim3[i,which(p.class==2&d.class==3)] /
                       n[which(p.class==2&d.class==3)])
}
post.mean.frame3 <- data.frame(means=c(m3.p1d1, m3.p2d1,
                                       m3.p1d2, m3.p2d2,
                                       m3.p1d3, m3.p2d3),
                               p.class=rep(rep(c("Number of Ponds 0-24",
                                                 "Number of Ponds 25-64"),
                                               each=4000), 3),
                               d.class=rep(c("Hatch Date 145-160",
                                             "Hatch Date 160-170",
                                             "Hatch Date 170-198"), each=8000))


ggplot(post.mean.frame3, aes(x=means)) + facet_grid(p.class ~ d.class) +
  geom_histogram(aes(y=..density..),
                 binwidth=0.01, col="blue", alpha=0.4, size=0.1, fill="blue") +
  theme_bw() +
  xlab("Average Proportion") + ylab("Density") +
  geom_vline(aes(xintercept=x), vline.frame)
```

```
zero.sim3 <- NULL
for(i in 1:4000){
  zero.sim3[i] <- sum(y.sim3[i,]==0)
}
zero.counts3 <- as.vector(table(zero.sim3))
zero.values3 <- dimnames(table(zero.sim3))$zero.sim3
zero.frame3 <- data.frame(y=zero.counts3, x=as.numeric(zero.values3))

all.sim3 <- NULL
for(i in 1:4000){
  all.sim3[i] <- sum(y.sim3[i,]==n)
}
all.counts3 <- as.vector(table(all.sim3))
all.values3 <- dimnames(table(all.sim3))$all.sim3
all.frame3 <- data.frame(y=all.counts3, x=as.numeric(all.values3))

ggplot(zero.frame3, aes(y=y, x=x)) + geom_point(col="blue") +
  geom_segment(aes(xend=x, yend=0), col="blue", lineend="round") +
  theme_bw() + xlab("Number of Broods") + ylab("Count") +
  ggtitle("Broods with Zero Surviving Ducklings") +
  theme(plot.title=element_text(face="plain")) +
  geom_vline(xintercept=sum(y==0), linetype="longdash")
ggplot(all.frame3, aes(y=y, x=x)) + geom_point(col="blue") +
```

```
  geom_segment(aes(xend=x, yend=0), col="blue", lineend="round") +
  theme_bw() + xlab("Number of Broods") + ylab("Count") +
  ggtitle("Broods with All Ducklings Surviving") +
  geom_vline(xintercept=sum(y==n), linetype="longdash")
```

```
early0.sim3 <- late0.sim3 <- NULL
for(i in 1:4000){
  early0.sim3[i] <- sum(y.sim3[i, which(hatchd < 180)] == 0)
  late0.sim3[i] <- sum(y.sim3[i, which(hatchd > 179)] == 0)
}
early0.counts3 <- as.vector(table(early0.sim3))
early0.values3 <- dimnames(table(early0.sim3))$early0.sim3
late0.counts3 <- as.vector(table(late0.sim3))
late0.values3 <- dimnames(table(late0.sim3))$late0.sim3
split0.frame3 <- data.frame(y=c(early0.counts3, late0.counts3),
                            x=c(as.numeric(early0.values3),
                                as.numeric(late0.values3)),
                            time=c(rep("Hatch Date 145-179",
                                       length(early0.counts3)),
                                   rep("Hatch Date 180-198",
                                       length(late0.counts3))))

ggplot(split0.frame3, aes(y=y, x=x)) + geom_point(col="blue") +
  geom_segment(aes(xend=x, yend=0), col="blue", lineend="round") +
  theme_bw() + xlab("Number of Broods") + ylab("Count") +
  facet_grid(~time, scales="free") +
  theme(plot.title=element_text(face="plain")) +
  geom_vline(aes(xintercept=x), linetype="longdash", data=vline.split)
```

4.5.4

```
resids3 <- extract(samps3, pars=c("resids"))$resids

for(i in 1:3){
  temp <- data.frame(resids=resids3[i, ],
                     nponds=dd.full$nponds,
                     hatchd=dd.full$hatchd)
  print(ggplot(temp, aes(x=nponds, y=resids)) + theme_bw() +
          geom_point(col="blue") + xlab("Number of Ponds") +
          ylab("Overall Pearson Residual") +
          theme(plot.title=element_text(face="plain")) +
          geom_hline(aes(yintercept=0)) + ylim(c(-2, 4)))
  print(ggplot(temp, aes(x=hatchd, y=resids)) + theme_bw() +
          geom_point(col="blue") + xlab("Hatch Date") +
          ylab("Overall Pearson Residual") +
          theme(plot.title=element_text(face="plain")) +
          geom_hline(aes(yintercept=0)) + ylim(c(-2, 4)))
}
```

```
post.z <- extract(samps3, pars=c("z"))$z
psi3 <- extract(samps3, pars=c("psi"))$psi
post.p3 <- extract(samps3, pars=c("p"))$p

for(i in 1:3){
  z.ind1 <- which(post.z[i, ]==1)
  resids1 <- (y[z.ind1] - n[z.ind1]*post.p3[i, z.ind1]) /
    (sqrt(n[z.ind1]*post.p3[i, z.ind1]*(1-post.p3[i, z.ind1])))
  temp.bin <- data.frame(resids=resids1,
                         nponds=dd.full$nponds[z.ind1],
                         hatchd=dd.full$hatchd[z.ind1])
  print(ggplot(temp.bin, aes(x=nponds, y=resids)) + theme_bw() +
          geom_point(col="blue") + xlab("Number of Ponds") +
          ylab("Binomial Pearson Residual") +
          theme(plot.title=element_text(face="plain")) +
          geom_hline(aes(yintercept=0)) + ylim(c(-3, 3)))
  print(ggplot(temp.bin, aes(x=hatchd, y=resids)) + theme_bw() +
          geom_point(col="blue") + xlab("Hatch Date") +
          ylab("Binomial Pearson Residual") +
          theme(plot.title=element_text(face="plain")) +
          geom_hline(aes(yintercept=0)) + ylim(c(-3, 3)))
}
```

```
post.psi3 <- extract(samps3, pars=c("psi"))$psi

np.breaks <- quantile(dd.full$nponds, seq(0, 1, 1/18))
np.breaks[1] <- np.breaks[1] - 0.1
np.breaks[19] <- np.breaks[19] + 0.1
np.groups <- cut(dd.full$nponds, np.breaks, labels=FALSE)

hd.breaks <- quantile(dd.full$hatchd, seq(0, 1, 1/18))
hd.breaks[1] <- hd.breaks[1] - 0.1
hd.breaks[19] <- hd.breaks[19] + 0.1
hd.breaks <- hd.breaks[-10]
hd.groups <- cut(dd.full$hatchd, hd.breaks, labels=FALSE)

np.means <- tapply(dd.full$nponds, np.groups, mean)
hd.means <- tapply(dd.full$hatchd, hd.groups, mean)

for(i in 1:3){
  resids2 <- (post.z[i, ] - post.psi3[i]) /
    (sqrt(post.psi3[i]*(1-post.psi3[i])))

  binned.np <- tapply(resids2, np.groups, mean)
  binned.hd <- tapply(resids2, hd.groups, mean)

  temp.bin1 <- data.frame(resids1=binned.np,
                          nponds=np.means)
  temp.bin2 <- data.frame(resids2=binned.hd,
                          hatchd=hd.means)
  print(ggplot(temp.bin1, aes(x=nponds, y=resids1)) + theme_bw() +
```

```
          geom_point(col="blue") + xlab("Number of Ponds") +
          ylab("Binned Bernoulli Pearson Residual") +
          theme(plot.title=element_text(face="plain")) +
          geom_hline(aes(yintercept=0)) + ylim(c(-2, 2)))
  print(ggplot(temp.bin2, aes(x=hatchd, y=resids2)) + theme_bw() +
          geom_point(col="blue") + xlab("Hatch Date") +
          ylab("Binned Bernoulli Pearson Residual") +
          theme(plot.title=element_text(face="plain")) +
          geom_hline(aes(yintercept=0)) + ylim(c(-2, 2)))
}
```

Zero-Inflated Binomial Model with Covariates for π 4.6.1

```
stan4 <- stan_model("zero_infl_cov.stan",
                    model_name="zero_infl_covs")
samps4 <- sampling(stan4, data=stan.dat)
```

```
table4 <- summary(samps4, pars=c("betas", "alphas"))$summary[, -c(9, 10)]
dimnames(table4)[[1]] <- c("$\\beta_0$", "$\\beta_1$(NP)", "$\\beta_2$(HD)",
                           "$\\beta_3$(HD$^2$)", "$\\beta_4$(NP$\\cdot$HD)",
                           "$\\beta_5$(NP$\\cdot$HD$^2$)", "$\\alpha_0$",
                           "$\\alpha_1$(NP)")
dimnames(table4)[[2]] <- c("Mean", "SE mean", "SD",
                           "2.5\\%", "25\\%", "50\\%", "75\\%", "97.5\\%")
print(xtable(table4, digits=2,
             caption='Summary of the posterior draws for each parameter
             from the Zero-Inflated Bayesian model with covariates'),
      caption.placement=getOption("xtable.caption.placement", "top"),
      sanitize.text.function=function(x) x )
```

```
plot(samps4, pars=c("betas", "alphas"), point_est="mean") +
  scale_y_continuous(labels=c(expression(alpha[1]),
                              expression(alpha[0]),
                              expression(beta[5]),
                              expression(beta[4]),
                              expression(beta[3]),
                              expression(beta[2]),
                              expression(beta[1]),
                              expression(beta[0])),
                     breaks=c(1:8))
```

4.6.2

```
alphas4 <- extract(samps4, pars=c("alphas"))$alphas
pi.fun <- function(ponds.c, alphas){
  ponds <- (ponds.c - mean(dd.full$nponds)) / sd(dd.full$nponds)
  return(plogis(cbind(1, ponds) %*% alphas[1:2]))
}
pi.mat <- apply(alphas4, 1,
                function(x){pi.fun(ponds.c, alphas=x)})
pi.means <- apply(pi.mat, 1, mean)
pi.lower <- apply(pi.mat, 1, quantile, probs=0.025)
pi.upper <- apply(pi.mat, 1, quantile, probs=0.975)

alpha.df <- data.frame(ponds=ponds.c, pi.mean=pi.means,
                       pi.low=pi.lower, pi.up=pi.upper)
ggplot(alpha.df, aes(x=ponds, y=pi.mean)) +
  geom_line(col="blue") + theme_bw() +
  geom_ribbon(aes(ymin=pi.low, ymax=pi.up),
              col="blue", alpha=0.2, linetype=0) +
  xlab("Number of Ponds") +
  ylab("Pr(No Catastrophic Event)")
```

```
betas4 <- extract(samps4, pars=c("betas"))$betas
z.mat4 <- apply(betas4, 1, function(x){outer(hd.c, ponds.c, z.fun1, beta=x)})
z.means4 <- apply(z.mat4, 1, mean)
z.means4 <- matrix(z.means4, nrow=54, ncol=65)

#Surface
par(mai=c(0.25, 0, 0.25, 0))
persp(hd.c, ponds.c, z.means4, theta=-30, phi=30, xlab="HD", ylab="NP", zlab="P",
      ltheta=120, shade=0.25, col="lightblue", lwd=0.25,
      main="", cex.lab=0.5, xaxs="i", zlim=c(0,1),
      ticktype="detailed", cex.axis=0.5)
mtext("Mean Posterior Surface")
```

```
z.lo4 <- apply(z.mat4, 1, quantile, probs=0.025)
z.lo4 <- matrix(z.lo4, nrow=54, ncol=65)
z.up4 <- apply(z.mat4, 1, quantile, probs=0.975)
z.up4 <- matrix(z.up4, nrow=54, ncol=65)

#lower
par(mai=c(0, 0.1, 0.25, 0.1))
persp(hd.c, ponds.c, z.lo4, theta=-30, phi=30, xlab="HD", ylab="NP", zlab="P",
      ltheta=120, shade=0.25, col="lightblue", lwd=0.25,
      main="", cex.lab=0.5, xaxs="i", zlim=c(0,1),
      ticktype="detailed", cex.axis=0.5)
mtext("Lower 95% PI Surface")

#upper
persp(hd.c, ponds.c, z.up4, theta=-30, phi=30, xlab="HD", ylab="NP", zlab="P",
      ltheta=120, shade=0.25, col="lightblue", lwd=0.25,
```

```
      main="", cex.lab=0.5, xaxs="i", zlim=c(0,1),
      ticktype="detailed", cex.axis=0.5)
mtext("Upper 95% PI Surface")
```

4.6.3

```
post.psi4 <- extract(samps4, pars=c("psi"))$psi
post.z4 <- extract(samps4, pars=c("z"))$z
post.p4 <- extract(samps4, pars=c("p"))$p

for(i in 1:3){
  resids4 <- (post.z4[i, ] - post.psi4[i, ]) /
    (sqrt(post.psi4[i]*(1-post.psi4[i, ])))

  binned.np2 <- tapply(resids4, np.groups, mean)
  binned.hd2 <- tapply(resids4, hd.groups, mean)

  temp.bin3 <- data.frame(resids3=binned.np2,
                          nponds=np.means)
  temp.bin4 <- data.frame(resids4=binned.hd2,
                          hatchd=hd.means)
  print(ggplot(temp.bin3, aes(x=nponds, y=resids3)) + theme_bw() +
          geom_point(col="blue") + xlab("Number of Ponds") +
          ylab("Binned Bernoulli Pearson Residual") +
          theme(plot.title=element_text(face="plain")) +
          geom_hline(aes(yintercept=0)) + ylim(c(-2, 2)))
  print(ggplot(temp.bin4, aes(x=hatchd, y=resids4)) + theme_bw() +
          geom_point(col="blue") + xlab("Hatch Date") +
          ylab("Binned Bernoulli Pearson Residual") +
          theme(plot.title=element_text(face="plain")) +
          geom_hline(aes(yintercept=0)) + ylim(c(-2, 2)))
}
```

Additional Inferences 5.1

```
z.fun3 <- function(hd.c, ponds.c, beta){
  hd <- (hd.c - mean(dd.full$hatchd)) / sd(dd.full$hatchd)
  ponds <- (ponds.c - mean(dd.full$nponds)) / sd(dd.full$nponds)
  return(plogis(cbind(1, ponds, hd, hd^2, ponds*hd, ponds*hd^2) %*% beta[1:6]) *
           plogis(beta[7] + beta[8]*ponds))
}

betas4_alpha <- cbind(betas4, extract(samps4, pars=c("alphas"))$alphas)

z.mat4_un <- apply(betas4_alpha, 1,
                   function(x){outer(hd.c, ponds.c, z.fun3, beta=x)})
z.means4_un <- apply(z.mat4_un, 1, mean)
z.means4_un <- matrix(z.means4_un, nrow=54, ncol=65)
```

```
#Surface
par(mai=c(0.25, 0, 0.25, 0))
persp(hd.c, ponds.c, z.means4_un, theta=-30, phi=30, xlab="HD", ylab="NP",
      zlab="P", ltheta=120, shade=0.25, col="lightblue", lwd=0.25,
      main="", cex.lab=0.5, xaxs="i", zlim=c(0,1),
      ticktype="detailed", cex.axis=0.5)
mtext("Mean Posterior Surface")
```

```
z.lo4_un <- apply(z.mat4_un, 1, quantile, probs=0.025)
z.lo4_un <- matrix(z.lo4_un, nrow=54, ncol=65)
z.up4_un <- apply(z.mat4_un, 1, quantile, probs=0.975)
z.up4_un <- matrix(z.up4_un, nrow=54, ncol=65)

#lower
par(mai=c(0, 0.1, 0.25, 0.1))
persp(hd.c, ponds.c, z.lo4_un, theta=-30, phi=30, xlab="HD", ylab="NP", zlab="P",
      ltheta=120, shade=0.25, col="lightblue", lwd=0.25,
      main="", cex.lab=0.5, xaxs="i", zlim=c(0,1),
      ticktype="detailed", cex.axis=0.5)
mtext("Lower 95% PI Surface")

#upper
persp(hd.c, ponds.c, z.up4_un, theta=-30, phi=30, xlab="HD", ylab="NP", zlab="P",
      ltheta=120, shade=0.25, col="lightblue", lwd=0.25,
      main="", cex.lab=0.5, xaxs="i", zlim=c(0,1),
      ticktype="detailed", cex.axis=0.5)
mtext("Upper 95% PI Surface")
```

```
opt.hd <- function(betas){
  ponds <- (c(18:64) - mean(dd.full$nponds)) / sd(dd.full$nponds)
  opt <- -(betas[3]+ponds*betas[5]) /
           (2*(betas[4] + betas[6]*ponds))
  return(opt*sd(dd.full$hatchd) + mean(dd.full$hatchd))
}
optimal.hd3 <- apply(betas4, 1, opt.hd)

optimal.hd3.mean <- apply(optimal.hd3, 1, mean)
optimal.hd3.low <- apply(optimal.hd3, 1, quantile, probs=0.025)
optimal.hd3.up <- apply(optimal.hd3, 1, quantile, probs=0.975)

opt.df <- data.frame(nponds=c(18:64),
                     mean=optimal.hd3.mean,
                     low=optimal.hd3.low,
                     upp=optimal.hd3.up)

ggplot(opt.df, aes(x=nponds, y=mean)) +
  geom_line(col="blue") + theme_bw() +
  geom_ribbon(aes(ymin=low, ymax=upp),
              col="blue", alpha=0.2, linetype=0) +
```

```
  xlab("Number of Ponds") +
  ylab("Optimal Hatch Date")
```

```
z.fun4 <- function(hd.c, ponds.c, beta){
  hd <- (hd.c - mean(dd.full$hatchd)) / sd(dd.full$hatchd)
  ponds <- (ponds.c - mean(dd.full$nponds)) / sd(dd.full$nponds)
  return((1-(1-plogis(cbind(1, ponds, hd, hd^2, ponds*hd, ponds*hd^2) %*%
                        beta[1:6]))^8) *
           plogis(beta[7] + beta[8]*ponds))
}

z.mat4_bs <- apply(betas4_alpha, 1,
                   function(x){outer(hd.c, ponds.c, z.fun4, beta=x)})
z.means4_bs <- apply(z.mat4_bs, 1, mean)
z.means4_bs <- matrix(z.means4_bs, nrow=54, ncol=65)

#Surface
par(mai=c(0.25, 0, 0.25, 0))
persp(hd.c, ponds.c, z.means4_bs, theta=-30, phi=30, xlab="HD", ylab="NP",
      zlab="P", ltheta=120, shade=0.25, col="lightblue", lwd=0.25,
      main="", cex.lab=0.5, xaxs="i", zlim=c(0,1),
      ticktype="detailed", cex.axis=0.5)
mtext("Mean Posterior Surface")
```

```
z.lo4_bs <- apply(z.mat4_bs, 1, quantile, probs=0.025)
z.lo4_bs <- matrix(z.lo4_bs, nrow=54, ncol=65)
z.up4_bs <- apply(z.mat4_bs, 1, quantile, probs=0.975)
z.up4_bs <- matrix(z.up4_bs, nrow=54, ncol=65)

#lower
par(mai=c(0, 0.1, 0.25, 0.1))
persp(hd.c, ponds.c, z.lo4_bs, theta=-30, phi=30, xlab="HD", ylab="NP", zlab="P",
      ltheta=120, shade=0.25, col="lightblue", lwd=0.25,
      main="", cex.lab=0.5, xaxs="i", zlim=c(0,1),
      ticktype="detailed", cex.axis=0.5)
mtext("Lower 95% PI Surface")

#upper
persp(hd.c, ponds.c, z.up4_bs, theta=-30, phi=30, xlab="HD", ylab="NP", zlab="P",
      ltheta=120, shade=0.25, col="lightblue", lwd=0.25,
      main="", cex.lab=0.5, xaxs="i", zlim=c(0,1),
      ticktype="detailed", cex.axis=0.5)
mtext("Upper 95% PI Surface")
```

Traceplots

```
traceplot(samps1, pars=c("betas"), ncol=1)
```

```
traceplot(samps2, pars=c("betas", "sigma_e"), ncol=1)
```

```
traceplot(samps3, pars=c("betas", "alpha"), ncol=1)
```

```
traceplot(samps4, pars=c("betas", "alphas"), ncol=1)
```

A.3 Stan Model Code

Bayesian Logistic Regression

```
data{
  int<lower=0> N;
  int<lower=0> count[N];
  int m[N];

  vector[N] ponds;
  vector[N] hatchd;
}

transformed data{
  vector[N] hatchd2;
  vector[N] ponds_hd;
  vector[N] ponds_hd2;

  hatchd2 <- hatchd .* hatchd;
  ponds_hd <- ponds .* hatchd;
  ponds_hd2 <- ponds .* hatchd2;
}

parameters{
  vector[6] betas;
}

transformed parameters{
  vector[N] logit_p;

  logit_p <- betas[1] + betas[2]*ponds + betas[3]*hatchd + betas[4]*hatchd2 +
    betas[5]*ponds_hd + betas[6]*ponds_hd2;
}

model{
  betas ~ normal(0, 30);

  count ~ binomial_logit(m, logit_p);
}

generated quantities{
  int<lower=0> y_rep[N];

  for(i in 1:N){
    y_rep[i] <- binomial_rng(m[i], inv_logit(logit_p[i]));
  }
}
```

Accounting for Overdispersion using Normal Errors

```
data{
  int<lower=0> N;
  int<lower=0> count[N];
  int m[N];

  vector[N] ponds;
  vector[N] hatchd;
}

transformed data{
  vector[N] hatchd2;
  vector[N] ponds_hd;
  vector[N] ponds_hd2;

  hatchd2 <- hatchd .* hatchd;
  ponds_hd <- ponds .* hatchd;
  ponds_hd2 <- ponds .* hatchd2;
}

parameters{
  vector[6] betas;
  vector[N] epsilon;
  real<lower=0> sigma_e;
}

transformed parameters{
  vector[N] logit_p;

  logit_p <- betas[1] + betas[2]*ponds + betas[3]*hatchd + betas[4]*hatchd2 +
    betas[5]*ponds_hd + betas[6]*ponds_hd2 + epsilon;
}

model{
  betas ~ normal(0, 30);
  sigma_e ~ cauchy(0, 2.5);

  epsilon ~ normal(0, sigma_e);

  count ~ binomial_logit(m, logit_p);
}

generated quantities{
  int<lower=0> y_rep[N];

  for(i in 1:N){
    y_rep[i] <- binomial_rng(m[i], inv_logit(logit_p[i]));
  }
}
```

Zero-Inflated Binomial Approach

```
data{
  int<lower=0> N;
  int<lower=0> count[N];
  int m[N];

  vector[N] ponds;
  vector[N] hatchd;
}

transformed data{
  vector[N] hatchd2;
  vector[N] ponds_hd;
  vector[N] ponds_hd2;

  hatchd2 <- hatchd .* hatchd;
  ponds_hd <- ponds .* hatchd;
  ponds_hd2 <- ponds .* hatchd2;
}

parameters{
  vector[6] betas;
  real alpha;
}

transformed parameters{
  vector[N] p;
  real psi;

  for(i in 1:N){
    p[i] <- inv_logit(betas[1] + betas[2]*ponds[i] + betas[3]*hatchd[i] +
      betas[4]*hatchd2[i] + betas[5]*ponds_hd[i] + betas[6]*ponds_hd2[i]);
  }

  psi <- inv_logit(alpha);
}

model{
  real log_psi;
  real log1m_psi;

  log_psi <- log(psi);
  log1m_psi <- log1m(psi);

  betas ~ normal(0, 30);
  alpha ~ normal(0, 30);

  for(i in 1:N){
    if(count[i] > 0)
      increment_log_prob(log_psi + binomial_log(count[i], m[i], p[i]));
    else
      increment_log_prob(log_sum_exp(log_psi + binomial_log(count[i], m[i], p[i]),
      log1m_psi));
```

```
  }
}

generated quantities{
  int<lower=0, upper=1> z[N];
  int<lower=0> y_rep[N];

  int<lower=0, upper=1> z2[N];
  int<lower=0> y_rep2[N];

  vector[N] resids;

  for(i in 1:N){
    real cond_psi;
    real y_exp;
    real y_var;

    if(count[i] > 0)
      cond_psi <- 1;
    else
      cond_psi <- psi*((1-p[i])^m[i]) / (psi*((1-p[i])^m[i]) + (1-psi));
    z[i] <- bernoulli_rng(cond_psi);
    y_rep[i] <- binomial_rng(m[i], z[i]*p[i]);

    z2[i] <- bernoulli_rng(psi);
    y_rep2[i] <- binomial_rng(m[i], z2[i]*p[i]);

    y_exp <- psi*m[i]*p[i];
    y_var <- psi*m[i]*p[i]*(1-p[i]*(1-m[i]*(1-psi)));
    resids[i] <- (count[i] - y_exp) / sqrt(y_var);
  }
}
```

Zero-Inflated Binomial Model with Covariates for π

```
data{
  int<lower=0> N;
  int<lower=0> count[N];
  int m[N];

  vector[N] ponds;
  vector[N] hatchd;
}

transformed data{
  vector[N] hatchd2;
  vector[N] ponds_hd;
  vector[N] ponds_hd2;

  hatchd2 <- hatchd .* hatchd;
  ponds_hd <- ponds .* hatchd;
  ponds_hd2 <- ponds .* hatchd2;
}

parameters{
  vector[6] betas;
  vector[2] alphas;
}

transformed parameters{
  vector[N] p;
  vector[N] psi;

  for(i in 1:N){
    p[i] <- inv_logit(betas[1] + betas[2]*ponds[i] + betas[3]*hatchd[i] +
      betas[4]*hatchd2[i] + betas[5]*ponds_hd[i] + betas[6]*ponds_hd2[i]);
    psi[i] <- inv_logit(alphas[1] + alphas[2]*ponds[i]);
  }
}

model{
  vector[N] log_psi;
  vector[N] log1m_psi;

  for(i in 1:N){
    log_psi[i] <- log(psi[i]);
    log1m_psi[i] <- log1m(psi[i]);
  }

  betas ~ normal(0, 30);
  alphas ~ normal(0, 30);

  for(i in 1:N){
    if(count[i] > 0)
      increment_log_prob(log_psi[i] + binomial_log(count[i], m[i], p[i]));
    else
      increment_log_prob(log_sum_exp(log_psi[i] + binomial_log(count[i], m[i], p[i]),
```

```
      log1m_psi[i]));
  }
}

generated quantities{
  int<lower=0, upper=1> z[N];
  int<lower=0> y_rep[N];

  int<lower=0, upper=1> z2[N];
  int<lower=0> y_rep2[N];

  vector[N] resids;

  for(i in 1:N){
    real cond_psi;
    real y_exp;
    real y_var;

    if(count[i] > 0)
      cond_psi <- 1;
    else
      cond_psi <- psi[i]*((1-p[i])^m[i]) / (psi[i]*((1-p[i])^m[i]) + (1-psi[i]));
    z[i] <- bernoulli_rng(cond_psi);
    y_rep[i] <- binomial_rng(m[i], z[i]*p[i]);

    z2[i] <- bernoulli_rng(psi[i]);
    y_rep2[i] <- binomial_rng(m[i], z2[i]*p[i]);

    y_exp <- psi[i]*m[i]*p[i];
    y_var <- psi[i]*m[i]*p[i]*(1-p[i]*(1-m[i]*(1-psi[i])));
    resids[i] <- (count[i] - y_exp) / sqrt(y_var);
  }
}
```