BAYESIAN IN-SERVICE FAILURE RATE MODELS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

By
Tolunay Alankaya
August 2022

Bayesian in-service failure rate models
By Tolunay Alankaya
August 2022

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Savaş Dayanık(Advisor)

Taghi Khaniyev

Barış Ata

Approved for the Graduate School of Engineering and Science:

Orhan Arıkan

Director of the Graduate School

ABSTRACT

BAYESIAN IN-SERVICE FAILURE RATE MODELS

Tolunay Alankaya
M.S. in Industrial Engineering
Advisor: Savaş Dayanık
August 2022

Predicting the number of appliance failures during service after sales is crucial for manufacturers to detect production errors and plan spare part inventories. We provide a two-phased Bayesian model that predicts the number of refrigerators that fail after sales. Thus the study focuses on both sales forecasting and failure detection. The two-phased Bayesian model is trained by the datasets provided by a leading durable home appliances company. The accuracy results show that one-level models are inferior to multi-level models when the data are sparse. We conclude that hierarchical Bayesian models are preferable since they can naturally capture the heterogeneity across all blends of attributes.

Keywords: Hierarchical Bayesian models, Hamiltonian Monte Carlo, sales forecasting, in-service failures.

ÖZET

BAYEZYEN SERVİS İÇİ ARIZA ORANI MODELLERİ

Tolunay Alankaya
Endüstri Mühendisliği, Yüksek Lisans
Tez Danışmanı: Savaş Dayanık
Ağustos 2022

Satış sonrası arızalanan cihazların sayısını tahmin etmek, üreticilerin üretim hatalarını tespit etmeleri ve yedek parça stoklarını planlamaları için çok önemlidir. Satıştan sonra arızalanan buzdolaplarının sayısını tahmin eden iki aşamalı bir Bayes modeli sunuyoruz. Bu nedenle çalışma hem satış tahmini hem de arızalı ürün sayısı tahmini üzerine odaklanmaktadır. İki aşamalı Bayes modeli, lider bir dayanıklı ev aletleri şirketi tarafından sağlanan veriler tarafından eğitilmiştir. Sonuçlar, veriler seyrek olduğunda tek seviyeli modellerin çok seviyeli modellerden daha düşük performans sergilediğini göstermektedir. Hiyerarşik Bayes modellerinin verideki heterojenliği yakalayabildikleri için tercih edilebilir olduğu sonucuna varıyoruz.

Anahtar sözcükler: Hiyerarşik Bayes modelleri, Hamiltoncu Monte Carlo, satış tahmini, satış sonrası arızalar.

Acknowledgement

First of, I would like to express my gratitude to my advisor Prof. Savaş Dayanık for his support and his belief in me during the last years of my undergraduate education and my graduate studies. I wouldn't have been on this path without his never-ending support. It has been a great honor to work under his guidance.

I also would like to thank Prof. Barış Ata and Asst. Prof. Taghi Khaniyev for agreeing to read and review my thesis and providing valuable comments and suggestions. I would also like to thank all the professors and staff in the Department of Industrial Engineering for their help and support.

My friends Deniz Akkaya, Eylül Kabakcı, and all my colleagues I could not list here deserve special thanks for their support.

Finally, I acknowledge the people who mean a lot to me, my mother and father. I am thankful for your selfless love, care, pain, and sacrifice.

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

In the past decade, the popularity of Bayesian models and sampling methods flourished. Many researchers emphasize the strength and advantages of the hierarchical structures that can be straightforwardly fitted using sampling methods [1]. The urgency of hierarchical models becomes more evident when the practitioners work with sparse and hierarchical data [2]. The ability of hierarchical structures to maintain stable results on sparse data makes them more appealing for any type of prediction problem, especially for sales forecasting. For large-scale sales forecasts of household appliances, data heterogeneity and sparsity can be a substantial concern. Similarly, this study encounters these problems and uses Bayesian modeling as a remedy.

In this thesis, we introduced a statistical model that predicts the number of failed refrigerators in service. The study uses the data sets provided by a leading home appliance producer. Two datasets include information on the number of sold (installed) refrigerators and the number of failed refrigerators in service. The model consists of two parts that serve for sales forecasting and failure detection. Both datasets challenge the prediction methods due to the heterogeneity and sparsity of refrigerator models. To be able to obtain stable and precise predictions, we focused on hierarchical Bayesian models. Throughout the thesis, the methods related to Bayesian models are explained, and results from models fitted to data

are discussed.

As a classical statistical workflow, we first introduced the exploratory analysis of the data in Chapter 2. Graphical tools are used for understanding the effects of covariates. This chapter serves as a guideline for our applications. In Chapter 3, the related literature to our study is presented. Those studies include sales forecasting methods both in and outside the domestic appliance industry. Most of those studies focus on sophisticated forecasting methods and compare them with more traditional ones. Similar types of approaches are used for detecting defective goods. Both types of studies utilize the time-series methods and enhance them with more recent tools such as Bayesian sampling or artificial neural networks. In Chapter 4, some properties and characteristics of discrete generalized linear models are explained. This chapter is essential for gaining insight into appropriate distributions for the type of data at hand. To make a clear picture of Bayesian models, Chapter 5 explains the ideas behind them and introduces their recent sampling and diagnostics methods. The structures and Bayesian diagnostics of the competing models for the sales forecasting and in-service failure predictions are presented in Chapters 6 and 7, respectively. Lastly, models are compared in Chapter 8, and final comments are made in Chapter 9.

Chapter 2

Problem Definition and Datasets

This thesis aims to predict the number of in-service refrigerator failures using two datasets provided by a leading durable home appliances company. The company has more than 2800 retail stores in Turkey; thus, the provided data are rich and suitable for machine/statistical learning applications. Both of the datasets were collected monthly by the company between January 2018 and December 2020. Although results and inferences from these datasets may not fully represent the post-pandemic time, the company can make more accurate predictions of future periods by training the provided models with more recent data. For estimating the number of in-service refrigerator failures, the number of sold (installed) refrigerators is needed. Although this information is presented in the given datasets, it can be used only to estimate the past number of in-service refrigerator failures. Thus it can only be served for validating the model's estimation for the past but not for predicting the future. So, the number of installed refrigerators should also be predicted for future periods.

For this purpose, a two-phased statistical learning model is constructed. In the first phase, a model is built to predict the number of installed refrigerators using the covariates included in the installation dataset. This dataset has 2,906,434 rows and 20 columns, where each row represents a unique refrigerator installation. After grouping the refrigerators according to their attributes and preprocessing

the data, a new installation dataset with 30,642 rows and 7 columns is obtained. The inventory information is combined with this dataset; then, each row has information on the number of installed and uninstalled refrigerators with refrigerator covariates in each month for at most 36 months after production. In the second phase, we aim to detect the number of in-service refrigerator failures using the same structure in the first phase. The failure data set includes the number of failed refrigerators in each of 36 months after installation and shares the same covariates. In both installation and failure data, zero counts were missing. After adding the information about non-defective refrigerators (zero counts), we obtained data with 32,929 observations. Although the size of the failure data is more extensive than the installation data, it is significantly less informative due to the large variety of refrigerator types. The percentage of zeros is 66% for the failure data, while it is just 11% for installation data. Covariates in the datasets are summarized in Table 2.2. In addition to covariates, each data set includes a response variable y and a size variable n. For installation data, y is the number of installed refrigerators ($y_{\text{installed}}$), and n is the number of refrigerators ready for installation (n_{produced}). Similarly, the response variable in the failure data is the number of in-service refrigerator failures (y_{failed}), and the size of the risk group is the number of installed refrigerators ($n_{\text{installed}}$). The first six rows of installation and in-service failure data are presented in Tables 2.3 and 2.4, respectively. Descriptions of all variables used in the thesis are presented in Table 2.1.

Table 2.1: Descriptions of variables used in the thesis

Variable Names	**Definitions**
InstMon	Installation month of a refrigerator
ProdMon	Production month of a refrigerator
AgeInstall	The time between production and installation of a refrigerator
AgeFail	The time between installation and breakdown of a refrigerator
AgeFailBinary	Whether the breakdown of a refrigerator occurs in the first month after installing
RModel	Refrigerator model
Cmodel	Compressor model of a refrigerator
Model	Combined refrigerator and compressor model
Installed	Number of installed refrigerator
Produced	Number of refrigerators ready for installation
Failed	Number of failed refrigerators
ZeroInstalled	Zero installation
ZeroFailed	Zero failure

Table 2.2: Description of covariates

Covariate	Description	Levels
ProdMon	Production month of the refrigerator	1,2,...,12
InstMon	Installation month of the refrigerator	1,2,...,12
RModel	Encoding of the model of the refrigerator	aa1, ab2, ba3, bb1, ba5, bb19, cb1, cc1, cc3, fa2, ıa1, ja2, bb18, da1, da2, da4, ba15, ba7, ba8, bb10, ab3, ba9, bb11, ea1, ea2, eb5, eb6, ec1, eb4, ee1, a3, cb2, ba14, ca2, ga1, fa1, ıa2, ıa10, ba1, ab1, fa3, gb1, ed3, ba12, cc2, bb20, bb8, bb13, ba10, eb3, eb2, ha1, cb4, ja1, ha3, ba2, eb1, ba4, ed2, ad1, ba11, db2, bb3, bb2, db1, bb22, ba6, bb9, bb21, ıa6, bb14, ıa11, bb15, ec3, bb16, ac2, bb17, a8, db3, a7, fa4, ıa9, bc, cb3, ba13, fb, ee2
CModel	Encoding of the compressor model of the refrigerator	1a, 1b, 2d, 3a, 3b, 4a, 4b, 4c, 5a, 6a, 6b, 8b, 2a, 2b, 8e, 1c, 8a, 6e, 3d, 3c, 2f, 6c, 7a, 8f, 1d, 8c, 3e, 2g, 6d
Age	The time between production and installation in installation dataset/ The time between installation and breakdown in the failure dataset	Continuous

Table 2.3: Installation data

Age	ProdMon	InstMon	CModel	RModel	Model	Installed	Produced
0	1	1	1a	aa1	aa1-1a	56	3666
1	1	2	1a	aa1	aa1-1a	196	3610
2	1	3	1a	aa1	aa1-1a	218	3414
3	1	4	1a	aa1	aa1-1a	179	3196
4	1	5	1a	aa1	aa1-1a	254	3017
5	1	6	1a	aa1	aa1-1a	351	2763

Table 2.4: In-service failure data

Age	ProdMon	InstMon	CModel	RModel	Model	Failed	Installed
0	1	1	1a	aa1	aa1-1a	0	56
1	1	2	1a	aa1	aa1-1a	1	196
2	1	3	1a	aa1	aa1-1a	2	218
3	1	4	1a	aa1	aa1-1a	1	179
4	1	5	1a	aa1	aa1-1a	1	254
5	1	6	1a	aa1	aa1-1a	3	351

2.1 Exploring Datasets

In this section, we explore data to understand the covariates' effects on the number of installations and in-service failures. The findings from this analysis are used as a guideline for building statistical models.

2.1.1 Exploring Effects of Covariates in Installed Refrigerator Data

Figure 2.1 shows the installation numbers according to refrigerators' installation months. Each point represents the installation counts for each row in the data. It can be seen that there is a seasonality in the data. The number of installed refrigerators increases between January and August and then decreases. There is a peak in August with some distinct values. The variation in counts for each installation month is quite high. One of the sources of this variation can be the refrigerator model since some models have low installation rates even in the summer months.

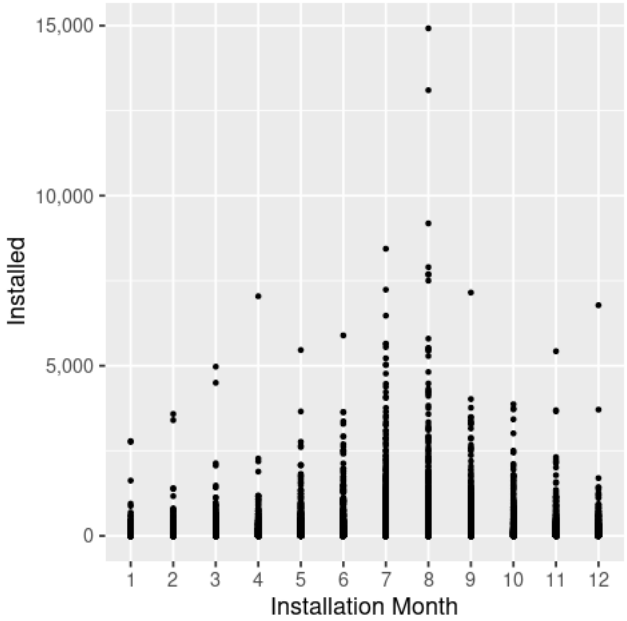


Figure 2.1: Installation month versus number of installed refrigerators

Figure 2.2 shows the installation numbers according to refrigerators' production months. There is also seasonality in the effect of production month, but its effect is weaker than installation month. It can be seen that number of installed refrigerators peaks before and beginning the summer. This observation might be expected since the company may increase its production rate to prepare for the excessive demands in the summer and then supply demand in the remainder of the year from stocks.

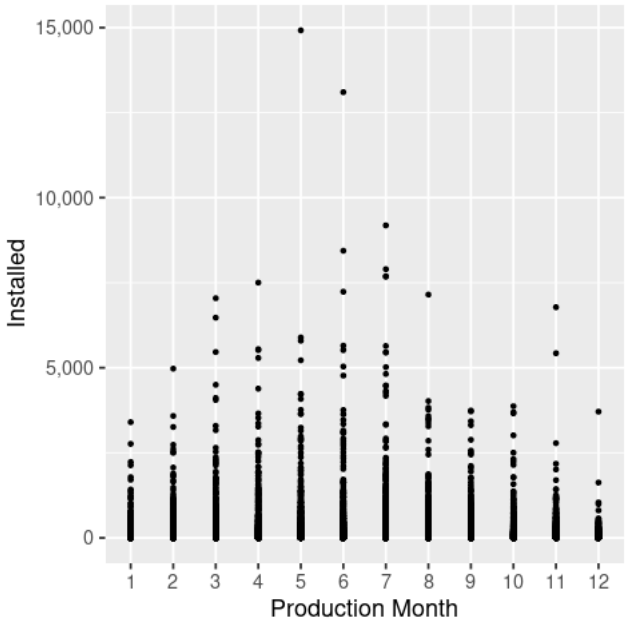


Figure 2.2: Production month versus number of installed refrigerators

From Figure 2.3, one can deduce that there is an inverse proportionality between age and the number of installed refrigerators. It can also be observed that the distribution is right skewed. The variation in installation numbers is quite large for small values of age, but it decreases as the age gets larger. A potential source of variations can be the refrigerator model, but Figure 2.8 indicates the variation in installation numbers are pretty high within refrigerator models. Another explanation for this variation can be the installation numbers of substitute appliances. Figure 2.3 indicates that the relation between installation numbers and the age might not be linear, and the peak occurs at an age larger than zero.

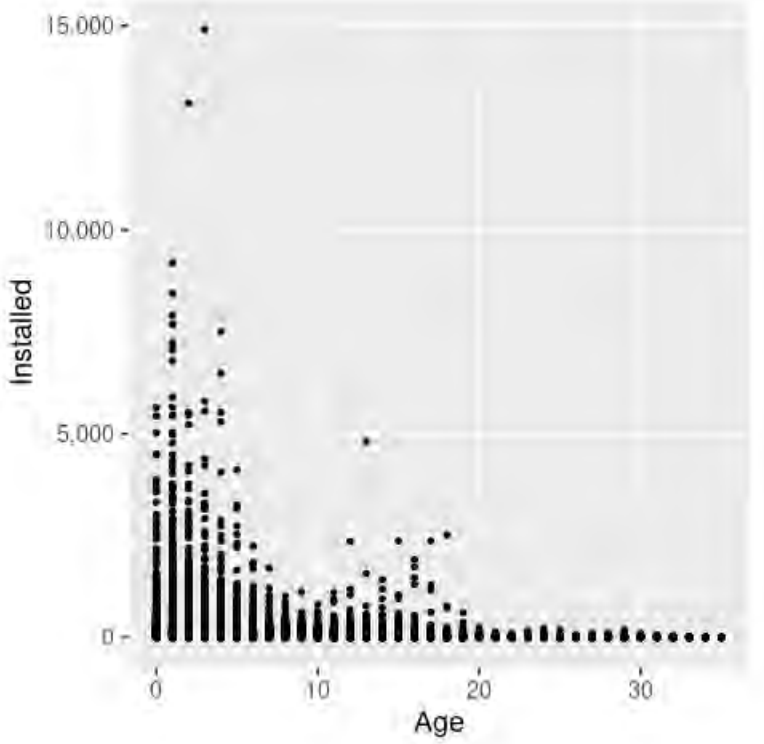


Figure 2.3: Age of the refrigerator versus number of installed refrigerators

Figures 2.4 and 2.5 shows the interaction effect of installation month and refrigerator model. The installation year of refrigerators is represented with different colors in those figures. One can observe that some models are only installed in 2018, meaning that new models substitute them. Also, some models have more recent installation years, and others are installed each year. Figures 2.4 and 2.5 also indicate that the seasonality effect on installation month highly changes according to refrigerator models. Models with low installation rates have a more negligible seasonality effect. On the other hand, installation numbers of more popular models dramatically increase during the summer season. A similar kind of behavior can be observed in Figures 2.6 and 2.7. Again, the seasonality effect of the production month is less significant for some models. Furthermore, these plots show that installation numbers highly depend on the refrigerator models. The popularities of refrigerators are pretty different, and the variance seems quite high. Some models are hardly ever sold. They can be relatively old models with less stock. Models with fewer installation numbers also have fewer observation

points. After all the covariates are added to the model, data can be sparse, and overfitting may occur. Thus rather than using the refrigerator model as a fixed effect, a grouping factor in a hierarchical model might produce better results.

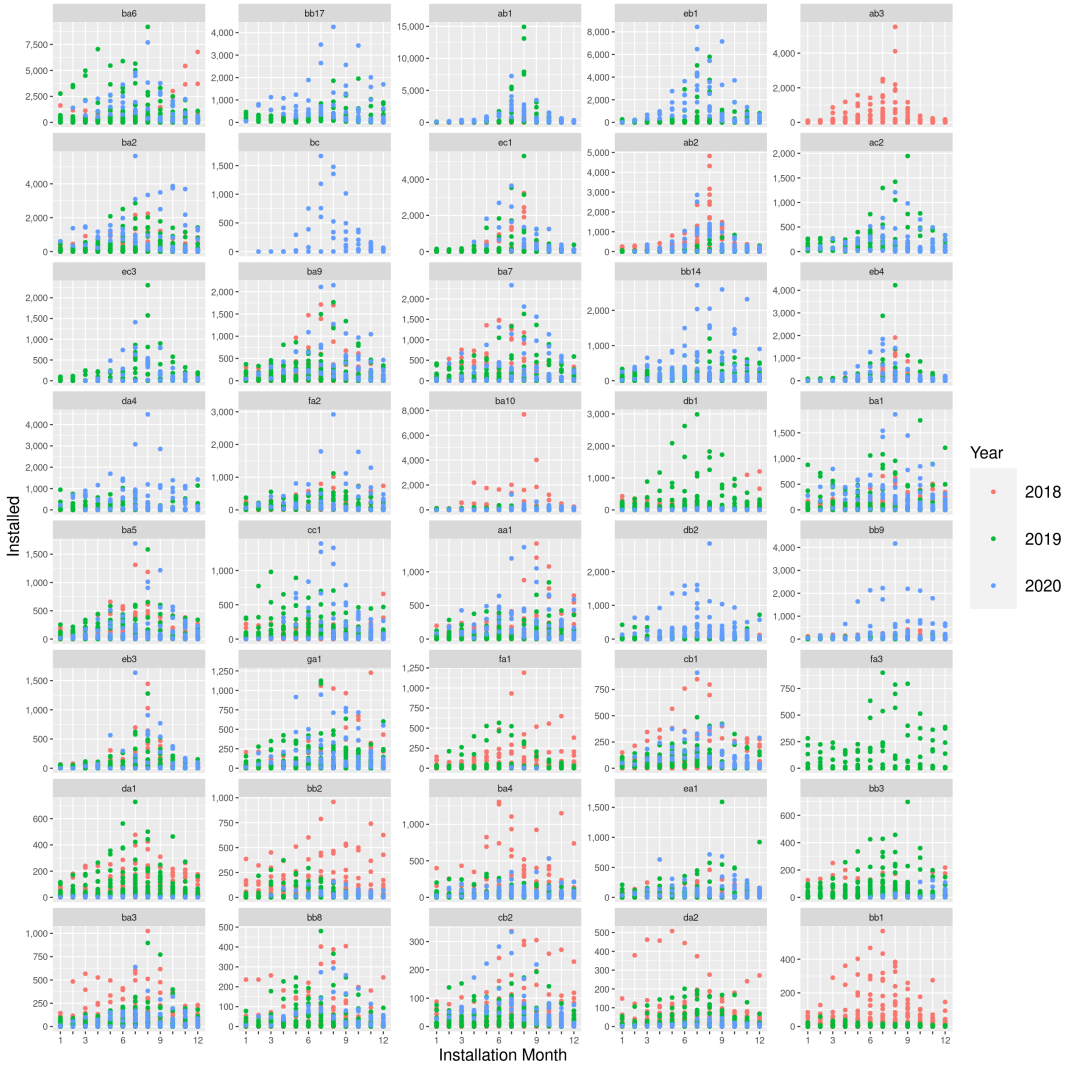


Figure 2.4: Interaction effect of installation month and refrigerator model

Figure 2.5: Interaction effect of installation month and refrigerator model

Figure 2.6: Interaction effect of production month and refrigerator model

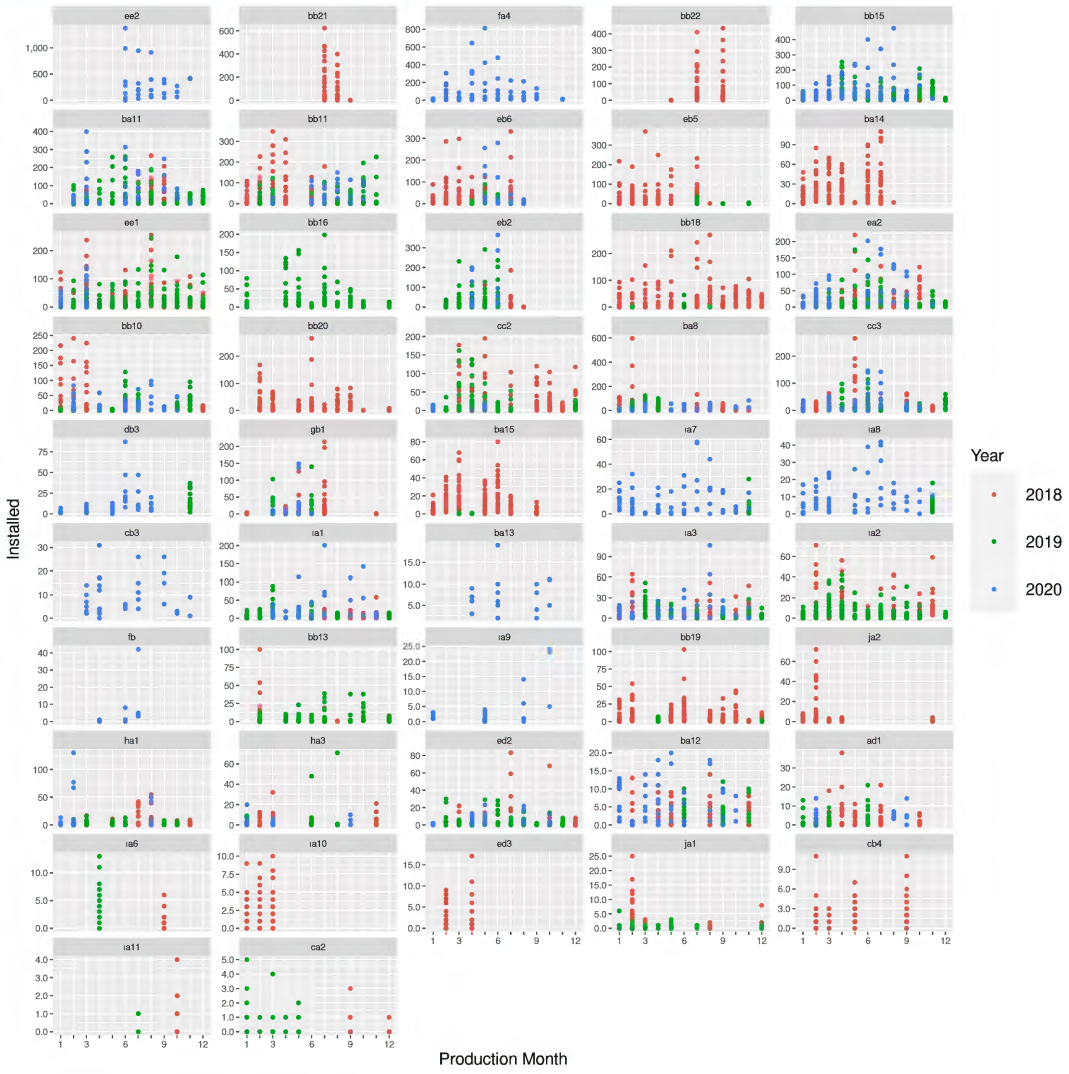


Figure 2.7: Interaction effect of production month and refrigerator model

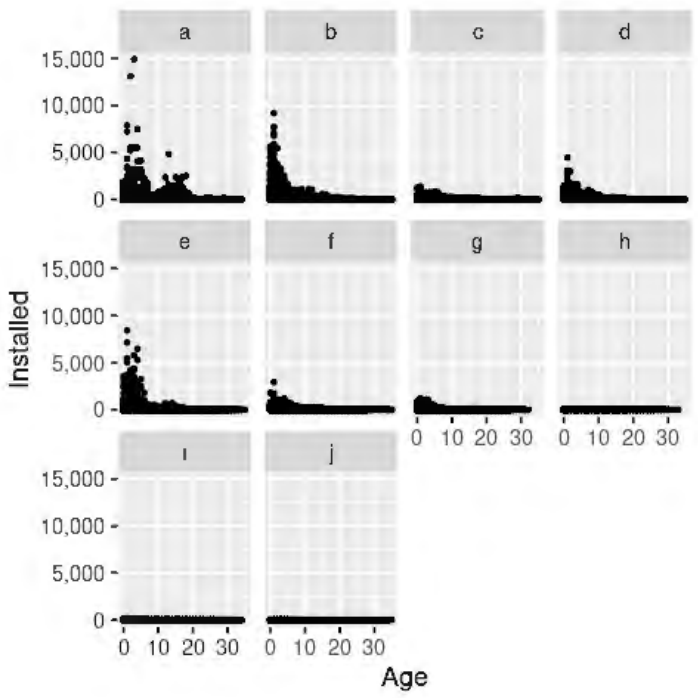


Figure 2.8: Interaction effect of age and refrigerator group

Figure 2.8 shows the interaction effect of age and the refrigerator group. Letters on top of the panels represent the first letter of the refrigerator model. The figure indicates that age interacts with the refrigerator model. For some models, such as the ones that start with "h," "ı," and "j," the effect of age is insignificant. For other models, the effect of age is visible, indicating that the hierarchical structure makes sense.

Figure 2.9 shows the relation between the compressor model and the installation month. It can be seen that the refrigerator compressor model interacts with the installation month. Similarly, Figure 2.10 shows the relation between the compressor model and the production month. The figure indicates that the compressor model also interacts with the production month. Thus, interaction terms can be added to the model, or the compressor model can be combined with the refrigerator model to create a unique model type for hierarchical structure.

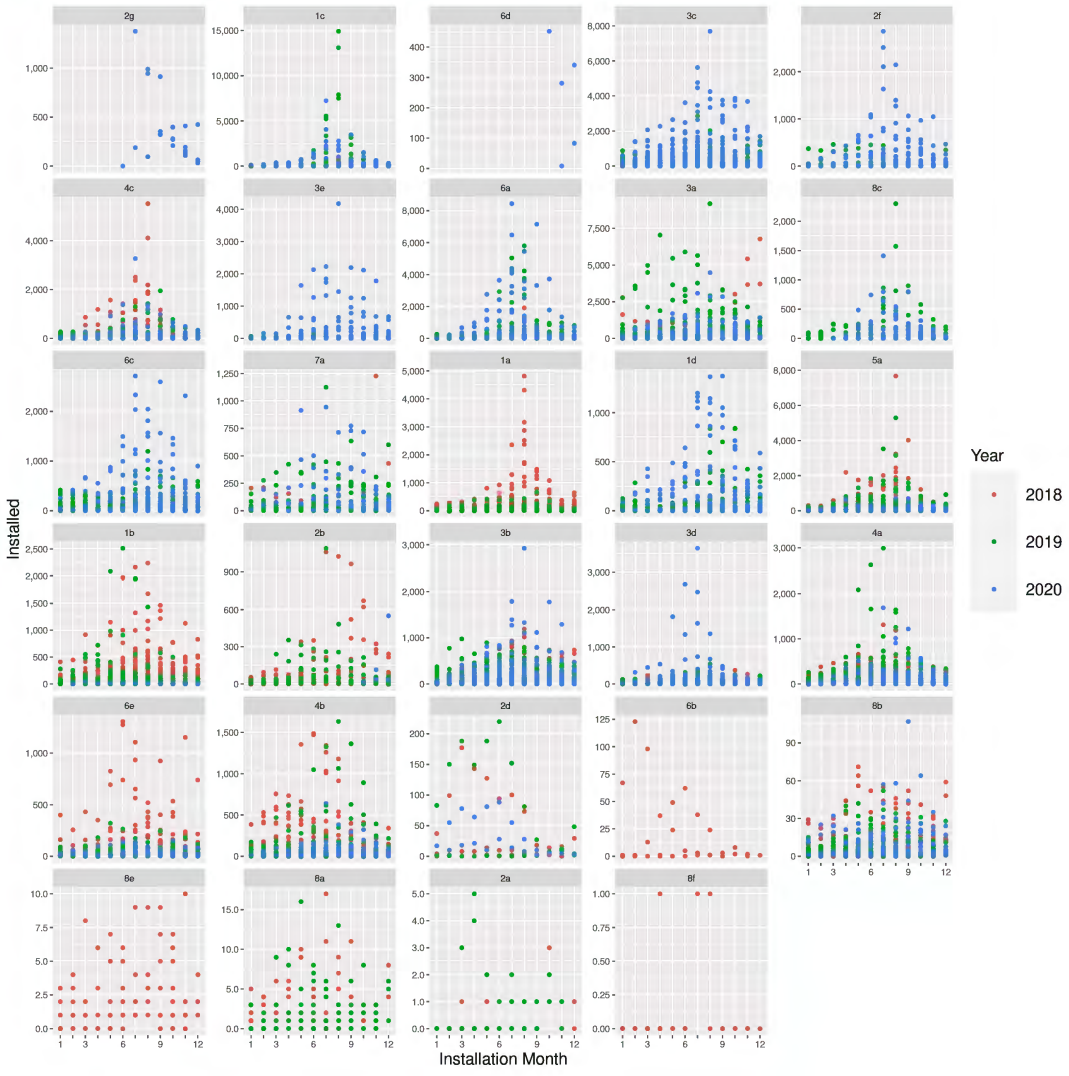


Figure 2.9: Interaction effect of installation month and compressor model

Figure 2.10: Interaction effect of production month and compressor model

2.1.2 Exploring Effects of Covariates on In-service refrigerator Failure Numbers

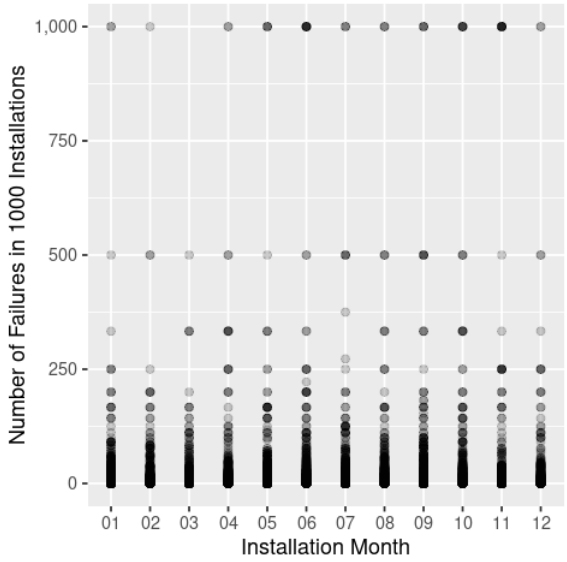


Figure 2.11: Installation month versus number of in-service refrigerator failures

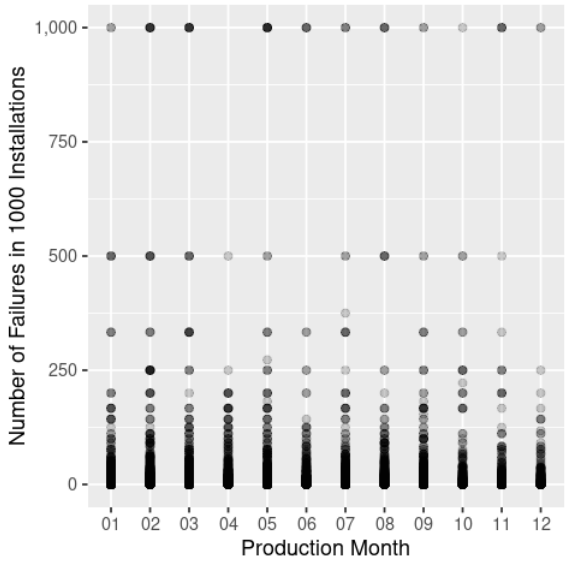


Figure 2.12: Production month versus number of in-service refrigerator failures

To take into account the installation numbers, we normalized the failures by installation numbers and used normalized failure rates in this section's figures. Figure 2.11 shows the relation between installation month and the number of in-service refrigerator failures. Transparent points in the figure indicate that the frequency of that point is low. One can deduce that the main effect of the installation month is not significant. Figure 2.12 shows the main effect of production month on in-service failure rates. Similar to the installation month, the production month doesn't have an impact on failure rates.

The age effect is presented in Figure 2.13. The figure indicates that the age effect on failure rate is nonlinear. One can notice that the number of failed refrigerators with age equal to one is distinctive. Thus, the age variable can turn into a binary variable that answers whether the refrigerator failed during the first month.

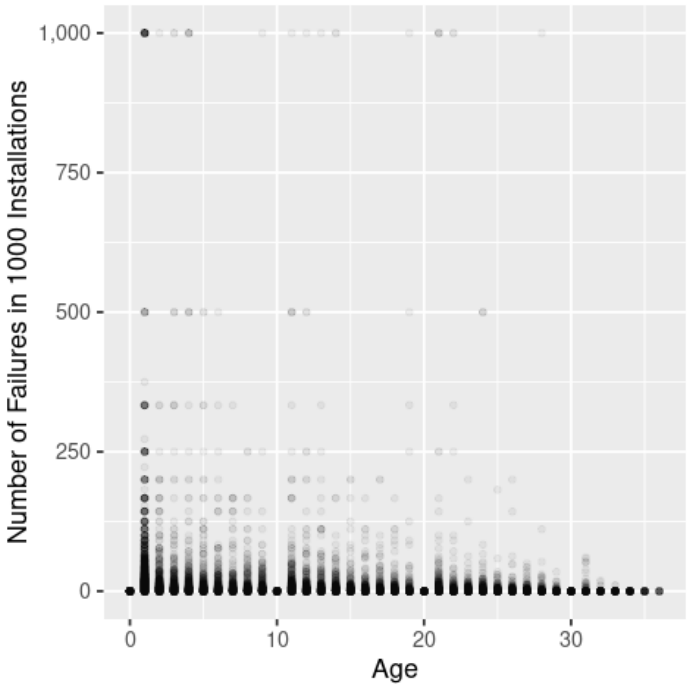


Figure 2.13: Age of the refrigerator versus number of in-service refrigerator failures

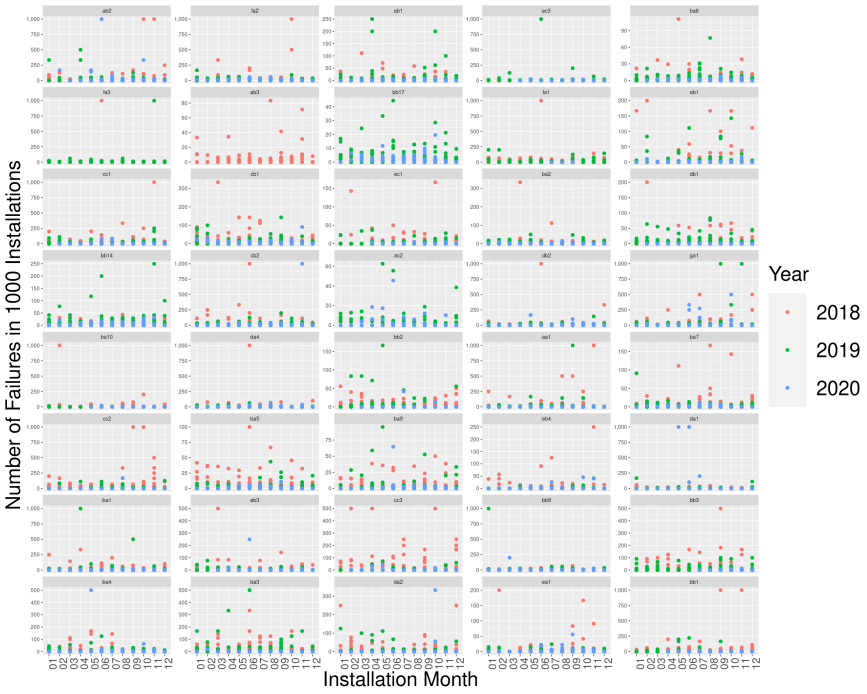


Figure 2.14: Interaction effect of installation month and refrigerator model

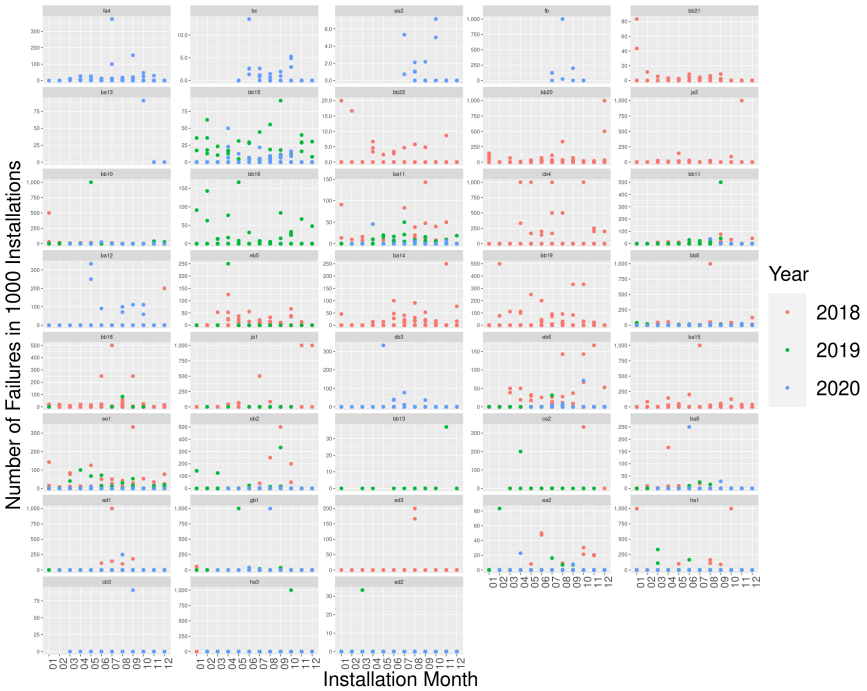


Figure 2.15: Interaction effect of installation month and refrigerator model

Figure 2.16: Interaction effect of production month and refrigerator model

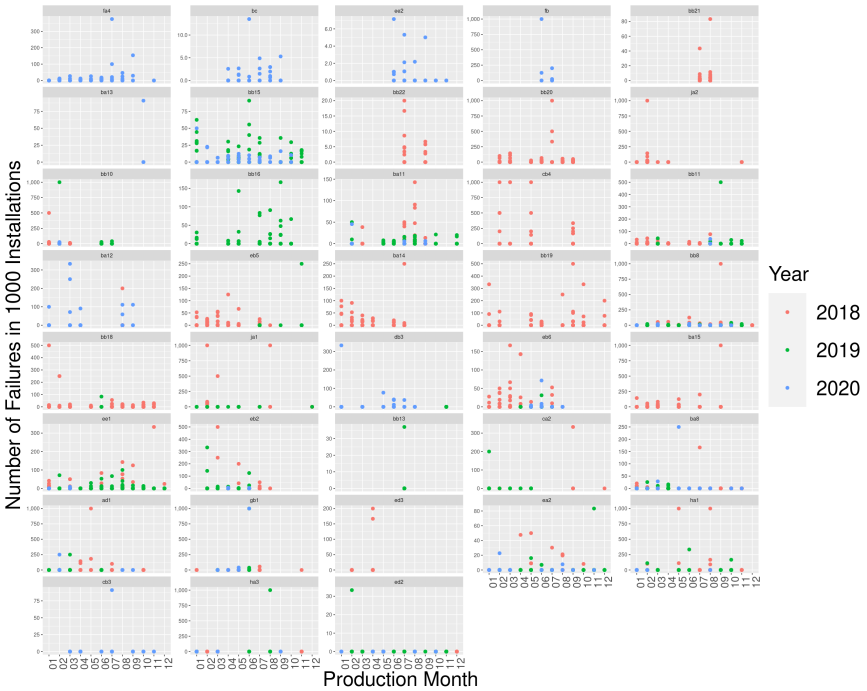


Figure 2.17: Interaction effect of production month and refrigerator model

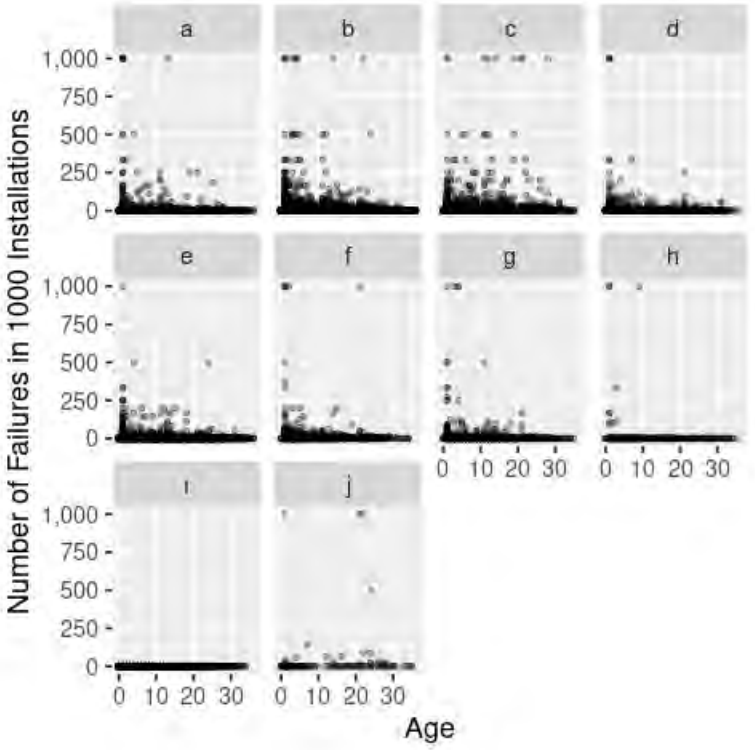


Figure 2.18: Interaction effect age and refrigerator group

Figures 2.14 and 2.15 indicate that for some models, the effect of installation month is more apparent. Unlike in Figure 2.11, failure rates are not fairly constant in those figures. Nonetheless, for most models, the installation month effect is weak. A hierarchical structure can be used for modeling this effect, or an interaction effect can be added to the models. However, hierarchical structures can provide a more robust solution since Figures 2.14, and 2.15 suggest that some models have considerably low data points. A similar type of interpretation of Figures 2.14 and 2.15 can also be made for Figures 2.16 and 2.17.

Figure 2.18 displays the interaction between age and refrigerator groups. One may notice that for some models, the effect of age is close to zero. On the contrary, the impact of the first month of the age is still apparent for some model groups. This observation also supports the need for hierarchical structure since the effect of age also depends on the refrigerator model.

Chapter 3

Literature Review

Sales forecasting is a widely studied topic by many researchers. Most applications are based on time-series analysis, conventional and recent methods, or count data models. The increasing popularity of Bayesian sampling methods and artificial neural networks reflects researchers' choice of forecasting tools, and these methods highly dominate this area. Recent studies usually use Bayesian sampling methods to estimate the parameters of state-space models or adapt time-series problems to artificial neural networks. The variety of techniques used to analyze the number of defective products is wide. In addition to the methods that are popular in sales forecasting, these area benefits from survival models.

Holt-Winters is a common prediction method in time-dependent problems. This method estimates the level, trend, and seasonality parameters of the time series. The improved version of the Holt-Winters method, the damped Holt-Winters model, fixes the unrealistic assumption that trends affect the whole horizon. A hybrid method proposed by Kotsialos, Papageorgiou, and Poulimenos [3] couples the Holt-Winters method with a feedforward multilayer neural network(FMNN). Simply, this method can be defined as using smoothing equations in Holt-Winters and producing the results with FMNN rather than an extrapolation equation. Eight variations of this model are presented in the paper. Results indicate that some FMNN-based models provide slightly better forecasting accuracies than the

damped Holt-Winters model.

The sales forecasts for household appliances are examined by Carmen [4] from a micro-economic point of view. The study uses cross-sectional data to include the effect of consumers' income, the price of substitute appliances, and the number of new family housing. The income elasticity of demand is considered for improving the forecasting results. After calculating the income elasticity, the author connects this information to the time series data and estimates the aforementioned covariates and time series parameters using a constrained regression model. The results are more promising than ordinary least square estimation.

In another forecasting study, Kolassa [5] focused on evaluation metrics of forecasting methods when the responses have a considerable portion of zeros. The study utilizes count data models such as Poisson and negative binomial regression. It also uses hybrid methods, including combining the count data models with the Croston model, bootstrapping, and dynamic programming. The author indicates some classical accuracy metrics, such as mean absolute deviation (MAD), mean absolute scaled error (MASE), and mean absolute percentage error (MAPE), are not suitable for assessing count data models. Additionally, the author emphasizes that mean squared error (MSE) can be misleading if the practitioner prioritizes increasing the accuracy of predicting zeros rather than large counts.

For forecasting the count valued time series data, Berry and West [6] proposed a novel state space model by combining dynamic generalized linear models with dynamic random effects. The study indicates that traditional time series methods like exponential smoothing, ARIMA, and linear state-space models are inappropriate when data includes too many zeros or low counts. The authors use a dynamic count mixture by expressing zeros and positive integers separately and enhance this model by adding a random effect to the state-space equation of the mean. The study indicates that the proposed model can handle the overdispersion problems in the data. Also, practitioners use the strength of the Bayesian approach by defining their model in such a way.

The zero-inflated Poisson regression model was introduced by Lambert [7] in

the context of detecting defects in manufacturing. The study argues zeros in the data should be expressed by another distribution rather than the Poisson distribution. The author proposed a discrete mixture model with covariates that enable the capability of representing zeros more accurately. The study shows that the Poisson distribution usually can not model the positive integers and zeros together even if the covariates carry enough information about the response. The author compares the zero-inflated model with the negative binomial model. The conclusion was when the zero portion is large enough, the zero-inflated Poisson model outperforms the overdispersed count models, such as the negative binomial model.

As in sales forecasting, predicting the defective number of goods is also possible using time-series methods. Reda, Challoob, and Omran [8] examined the defective percentages in a laboratory. The authors compare standard time series methods like exponential smoothing, moving average, and least square estimation. In another study, Wang, Ni, and Wang [9] used more advanced techniques and analyzed the defects in train wheels. The authors proposed a Bayesian state-space model and also provided an outlier detection method based on the Bayes factor.

The survival analysis methods are also used for examining defectiveness. These types of hazard rate models are quite popular in the health and pharmacy industry. Gao, Duan, and Rui [10] investigate the effect of social media on pharmacy product recall using a discrete-time survival model. The study models hazard rates using lagged covariates and compare the results of two different link functions. In another study, Reefhuis [11] use the Bayesian approach to estimate the log odds ratio of congenital disabilities according to mothers' characteristics and external factors such as smoking .

Chapter 4

Modeling Count Data

In this chapter of the thesis, some properties and characterizations of the discrete generalized linear models (GLMs) used in modeling count data are explained. Generalized linear models (GLMs) are assumed to belong to a general family of distributions called the exponential dispersion model family (EDMs) [12]. Discrete distributions in the EDM family have the probability mass function of the following form

$$\mathcal{P}(y;\theta,\phi) = a(y,\phi)\exp\left\{\frac{y\theta-\kappa(\theta)}{\phi}\right\}, \tag{4.1}$$

where

- θ is the canonical parameter characterizes the mean function μ,
- $\phi >0$ is the dispersion parameter that scales the variance function of y,
- $\kappa(\theta)$ is the cumulant function that defines the moments of the distribution,
- $a(y;\phi)$ is the normalizing function ensuring that $\sum_y \mathcal{P}(y;\theta,\phi) = 1$.

The response variable y has a distribution with mean μ and dispersion parameter ϕ. Most of the time, the variance assumption on y does not hold and leads to an overdispersion problem. As the name refers, overdispersion means that variance in the data is larger than the theoretical variance. To address overdispersion

mixture models are proposed. Using maximum likelihood estimation (MLE) to estimate the regression parameters can be challenging when mixture models do not appear as known distribution in the EDM family. On the other hand, some Bayesian sampling methods are quite flexible and can fit mixture models even if the priors are not conjugate priors of the observation distribution.

4.1 Binomial Regression

In the context of the first part of the defined problem, the number of installed refrigerators represented as $v_{\text{installed}}$ has a binomial distribution with the size n_{produced}; namely, the number of manufactured refrigerators, and installation probability $\mu_{\text{installed}}$. Similarly, the second part of the problem modeled as $v_{\text{defective}} \sim Bin\left(n_{\text{installed}}, \mu_{\text{defective}}\right)$, where $n_{\text{installed}}$ is the number of installed refrigerators and $\mu_{\text{defective}}$ is the probability that a refrigerator will fail in service. Using a response variable y as a success ratio, explicitly $y = \frac{v}{n}$, probability mass functions in both problems have the following form

$$\begin{aligned} \mathcal{P}\left(y; \mu, n\right) &= \binom{n}{ny} \mu^{ny} \left(1-\mu\right)^{n(1-y)}, \\ &= \binom{n}{ny} \exp\left[\frac{y \log\left(\frac{\mu}{1-\mu}\right) + \log\left(1-\mu\right)}{1/n}\right]. \end{aligned} \tag{4.2}$$

Here canonical parameter suggests the logit function as the natural link function of binomial regression. Denoting the predictors as X and their regression coefficients as β, the probability of an event being occurring is calculated using $\log \frac{\mu}{1-\mu} = X\beta$.

The r^{th} cumulant for an EDM equals

$$\kappa_r = \phi^{r-1} \frac{d^r \kappa\left(\theta\right)}{d\theta^r}. \tag{4.3}$$

Equation 4.2 shows that $\phi = \frac{1}{n}$, and $\kappa\left(\theta\right) = \log\left(1-\mu\right)$. Then, using equation 4.3 mean and variance of binomial regression can be defined as

$$E\left(y\right) = \mu, \qquad Var\left(y\right) = \frac{\mu\left(1-\mu\right)}{n}. \tag{4.4}$$

The theoretical variance of the response variable can be extremely small when the n is large enough; then, it might not match the variance in the data. Mixture models that allow overdispersion can be used as a solution. For scaling the variance of the binomial distribution, the probability p of an event is defined as a beta distributed random variable. By setting $\log \frac{\mu}{1-\mu} = X\beta$, the number of successive trials v and success probability p have the following distributions.

$$v \sim Bin\left(n, p\right) \qquad p \sim Beta\left(\frac{\mu}{\alpha}, \frac{1-\mu}{\alpha}\right) \tag{4.5}$$

The resultant mixture model is denoted as Beta-binomial distribution, and it does not belong to the EDM family. So, Equation 4.3 can not be used for calculating the variance of success ratio $y = \frac{v}{n}$. But the variance of y can be calculated using the variance of beta distribution and the law of the total variance [13]

$$E\left(y\right) = \mu, \qquad Var\left(y\right) = \frac{\mu\left(1-\mu\right)}{n}\left(1 + \frac{n-1}{\alpha+1}\right). \tag{4.6}$$

Equation 4.6 shows that the beta-binomial distribution has the same mean as the binomial distribution but a scaled version of its variance. Inverse-overdispersion parameter α characterizes the behavior of the distribution. When it goes to infinity, the variance will be equal to the variance of the binomial distribution. So, the beta-binomial distribution behaves similarly to the binomial distribution for the larger α values. Both regression parameters and α can be calculated using Bayesian sampling methods.

$$\lim_{n\to\infty} BetaBin\left(n, \mu, \alpha\right) \sim NB\left(\mu, \alpha\right) \tag{4.7}$$

An interesting aspect of the beta-binomial distribution is shown in Equation 4.7. The indication is limiting behavior of the beta-binomial distribution is analogous to the limiting behavior of the binomial distribution since the overdispersed version of the Poisson distribution is equivalent to the negative-binomial distribution (NB).

4.2 Poisson Regression

One of the Poisson distribution characterizations is obtained by taking the limit of trial size in the Binomial distribution. The law of rare events states that the number of successive trials approximately has Poisson distribution if the number of trials is large enough [14]. This statement makes the Poisson regression model suitable for the used dataset since the $n_{produced}$ and $n_{installed}$ are quite large. By denoting the response variable as count y, Poisson distribution has the probability function

$$\mathcal{P}(y;\mu) = \frac{\exp(-\mu)\,\mu^y}{y!} = \frac{1}{y!}\exp(y\log\mu - \mu), \tag{4.8}$$

where μ is the mean of the distribution. Here $\theta = \log\mu$, $\kappa(\theta) = \mu$, and $\phi = 1$. Canonical parameter suggests that the natural link function of the Poisson distribution is the logarithm function, and regression terms can be added using it (i.e.,$\log\mu = X\beta + \log(n)$). Here n is an offset and the general notation of the number of trials. Using the information on parameters and Equation 4.3 mean and variance of the Poisson distribution calculated as

$$E(y) = Var(y) = \mu, \tag{4.9}$$

which portrays an unrealistic assumption on mean and variance by considering them equal. Consequently, overdispersion is so common in Poisson regression models. A well-known mixture model is the Poisson-gamma mixture and achieved by using a gamma random variable for the mean of the Poisson distribution. Formally, let $y \sim \text{Pois}(\theta)$, $u \sim \text{gamma}(\alpha,\alpha)$,and $\theta = \mu u$, then the mixture model has the probability function

$$\begin{aligned}
\mathcal{P}(y;\mu,\alpha) &= \int f_{poisson}(y \mid \theta)\, f_{gamma}(\theta \mid \alpha,\mu)\, d\theta, \\
&= \int \frac{e^{-\theta}\theta^y}{y!}\frac{\left(\frac{\alpha}{\mu}\right)^{\alpha}}{\Gamma(\alpha)}\theta^{\alpha-1}e^{-\frac{\alpha\theta}{\mu}}d\theta, \\
&= \frac{\Gamma(\alpha+y)}{\Gamma(\alpha)\Gamma(y+1)}\left(\frac{\alpha}{\alpha+\mu}\right)^{\alpha}\left(\frac{\mu}{\alpha+\mu}\right)^{y}, \\
&= \frac{\Gamma(\alpha+y)}{\Gamma(\alpha)\Gamma(y+1)}\exp\left\{\alpha\log\left(\frac{\alpha}{\alpha+\mu}\right) + y\log\left(\frac{\mu}{\alpha+\mu}\right)\right\}.
\end{aligned} \tag{4.10}$$

The Poisson-gamma mixture is a characterization of the negative binomial distribution. The negative binomial distribution is a member of the EDM family with dispersion ϕ equals 1 if α is known, and the desired form is presented in the last line of Equation 4.10. Using the properties of EDM, the mean and variance of the negative binomial model calculated as

$$E(y) = \mu, \qquad Var(y) = \mu + \frac{\mu^2}{\alpha}. \tag{4.11}$$

As in the beta-binomial model, α is the inverse-overdispersion parameter and can be estimated with profile MLE or Bayesian sampling methods. Analogous to the limiting behavior of the beta-binomial model, the negative binomial model converges to the Poisson model as α goes to infinity.

4.3 Multi-Index Count Data Models

Like in the used dataset, generally, real-life datasets have excessive zeros than the expectation of the mentioned count data models. Multi-index models are offered to overcome this problem. Multi-index models have conditional mean functions, and the overall mean is not the main interest. Rather mean effect decomposed into an effect at the intensive margin and an effect at the extensive margin [15]. Hurdle models, also known as two-part models, specify different models for zero counts and positive integer counts [14], and they also address the misspecification caused by overdispersion [16]. The conditional probability function of familiar distributions like the Poisson distribution or negative binomial distribution can be used for modeling the positive integers. For modeling the remaining part, the response variable is considered as a binary outcome that equals zero or not. Since counts are conditioned on being larger than zero, this model uses a hurdle at zero. Any non-negative integer can be used as a hurdle, and appropriate distributions can be used for modeling two parts. The probability function of zero hurdle Poisson regression has the form

$$\mathcal{P}(y; \mu, \pi) = \begin{cases} \pi, & \text{if } y = 0, \\ (1-\pi)\,\frac{1}{1-\exp(-\mu)}\,\frac{\exp(-\mu)\mu^y}{y!}, & \text{if } y > 0, \end{cases} \tag{4.12}$$

for some $0 < \pi < 1$, $\mu > 0$. Here parameters π and μ can be calculated using regressors and appropriate link functions. Specifically, $\text{logit}(\pi) = X'\beta'$ and $\log(\mu) = X\beta + \log(n)$. In the zero hurdle models, the expected number of zeros exactly matches the number of zeros in the data. Thus it can handle the scenarios where data has too few zeros. This ability makes hurdle models more flexible compared to the second type of multi-index model, namely, zero-inflated models. Although hurdle models seem favorable over zero-inflated models, their performance highly depends on the data. The hurdle model poorly performs when the true data generating process is a zero-inflated distribution [17].

The idea in zero-inflated count models is adding extra zeros with some probability π. So, there are two types of sources for zeros in the zero-inflated models, structural and sampling zeros [17]. In the context of the defined problem, refrigerators that are produced but not sent to retail stores are included in the structural zero sources. On the other hand, refrigerators that are sent but not sold are included in the sampling zero sources. By using a similar setting used in the hurdle model, the probability function of the zero-inflated model can be defined as

$$\mathcal{P}(y;\mu,\pi) = \begin{cases} \pi + (1-\pi)\left(1 - \exp\{-\mu\}\right), & \text{if } y = 0, \\ (1-\pi)\,\frac{\exp(-\mu)\mu^y}{y!}, & \text{if } y > 0. \end{cases} \tag{4.13}$$

For both models, the formulation of the probability function can be redefined using negative-binomial distribution. Again, parameters in these models can be calculated using MLE or Bayesian sampling methods.

Chapter 5

Bayesian Models

The idea behind the Bayesian approach is to estimate the unknown model parameters $\theta = (\theta_1, \theta_2, \ldots, \theta_m)$ by updating the knowledge about them according to observations $y = (y_1, y_2, \ldots, y_n)$ and obtaining posterior densities $p(\theta \mid y)$. Using the Bayes formula, posterior densities are calculated with

$$p(\theta \mid y) = \frac{p(y \mid \theta)\, p(\theta)}{p(y)}. \tag{5.1}$$

Here $p(y \mid \theta)$ is the likelihood of y given model parameters θ. The frequentist approach aims at maximizing the likelihood by changing the model parameters, but the Bayesian approach focuses on finding the left-hand side of Equation 5.1. The likelihood is multiplied with prior $p(\theta)$, which can be considered as including the knowledge about model parameters or adding uncertainity to the model [18]. These interpretations of the prior depend on the choice of the prior distribution. Some choices of distributions are considered informative priors and used when practitioners want to dominate information that come from data with their expectations on parameters. In theory, priors can be non-informative and used for adding uncertainty. On the other hand, the effect of the prior depends on the context of the likelihood, and even non-informative priors can have substantial and unpleasant effects. Thus, weakly informative priors became more popular since they allow the data to inform while eliminating the unlikely values on parameters [19]. Nevertheless, prior distributions or hyperparameters of the distributions

need to be carefully selected using prior predictive checks and simulations.

In Equation 5.1, $p(y)$ is the marginal density of y, which can be calculated by integrating the numerator:

$$p(y) = \int p(y \mid \theta)\, p(\theta)\, d\theta, \tag{5.2}$$

where $p(y)$ can be considered as a normalizing factor that ensures posterior density adds up to 1 [20]. Thus, without calculating the marginal density, posteriors can be found using the fact that the posterior is proportional to the likelihood times prior:

$$p(\theta \mid y) \propto p(y \mid \theta)\, p(\theta)\,. \tag{5.3}$$

Due to its nature, the Bayesian approach allows complex models such as hierarchical (also known as multilevel) models. A hierarchical model groups the data by a factor and adjusts the model parameters accordingly. This type of parameter estimation is called partial-pooling, which is an intermediate estimation type between complete and no-pooling. No-pooling is equivalent to the fixed effect model, which estimates the parameters using the sub-data obtained by dividing data into the observations with the specific factor levels. No-pooling can lead to over-fitting when the sample size of the sub-data is insufficient. On the other hand, complete pooling uses the whole data when predicting the response, so it is basically the sample mean of the response. Obviously, complete pooling is not desirable since covariates are discarded. Partial-pooling takes advantage of both complete-and no-pooling. The intuitive idea behind partial-pooling is shrinking model parameters towards their means. The amount of pulling increases as the sample size of the sub-data decreases. In the context of refrigerator data, let us assume that the predictor is installation month, and the grouping factor is the refrigerator model. Each level of installation month has a mean random effect that can be calculated by discarding the refrigerator model. When the grouping factor is included, a random effect of installation month can be different than its mean. The difference will be less if the number of observations for a refrigerator model is small and vice versa. This model can be constructed by using the installation month random effect as the parameter of the refrigerator-specific installation month effect.

Formally, let α be the likelihood parameter, β be the group-specific effect, and γ be the group-specific parameter (e.g., installation month effect). Then the posterior density takes the form [21]

$$p\left(\beta, \alpha, \gamma \mid y\right) = \frac{p\left(y \mid \beta, \alpha\right) p\left(\beta \mid \gamma\right) p\left(\gamma\right) p\left(\alpha\right)}{p\left(y\right)}. \tag{5.4}$$

In addition to likelihood and priors, Equation 5.4 has the conditional density of group-specific effects. Here β and γ are vectors, and priors of γ can be written as multiplication due to the independence assumption. The conditional density of β is usually selected as multivariate distribution to express the correlation between covariates. This makes multilevel models even more appealing.

5.1 Introduction to Bayesian Sampling

When the priors are selected as conjugate priors (i.e. prior and posterior distributions belong to the same distribution family), an analytical solution of posterior distribution can be possible. Also, posteriors can be approximated using approximation techniques such as Laplace approximation [20]. When the parameter space is large, or the model is complex due to its hierarchical structure, calculating marginal densities with former approaches can be difficult or even impossible. In that case, Markov Chain Monte Carlo (MCMC) sampling methods offer a solution to obtain posterior densities [21].

MCMC explores the parameter space in a Markovian way by using transition probabilities. In each iteration, value of a parameter changes or stays the same according to its transition probability to the candidate value. Formally let $\theta^{(t)}$ be the sampled value at iteration t. Then transition probability can be defined as

$$K\left(\theta^{(t)} \mid \theta^{(0)}, \theta^{(1)}, \ldots, \theta^{(t-1)}\right) = K\left(\theta^{(t)} \mid \theta^{(t-1)}\right). \tag{5.5}$$

With an appropriate acceptance-rejecting rule, transition distribution K ensures that the sequence of sampled parameter values is equivalent to $p\left(\theta \mid y\right)$. After some iterations, B, transition probabilities reach their stationary distribution.

Then mean and variance of θ can be calculated empirically [20] by

$$\begin{aligned} E(\theta) &= \sum_{t=B+1}^{T} \theta^{(t)} / (T-B), \\ V(\theta) &= \sum_{t=B+1}^{T} \left(\theta^{(t)} - E(\theta)\right)^2 / (T-B). \end{aligned} \tag{5.6}$$

5.1.1 Metropolis Sampling

Metropolis sampling proposes a method that randomly walks around the parameter space using proposal distribution $p\left(\theta_{cand} \mid \theta^{(t)}\right)$. Proposal distribution can be selected as any symmetrical distribution. In each iteration, θ_{cand} is generated from proposal density (e.g., $\theta_{cand} \sim N(\theta^{(t)}, \sigma^2)$), then its acceptance probability is calculated by

$$p^{(t)} = \min\left(1, \frac{p(\theta_{cand} \mid y)}{p(\theta^{(t)} \mid y)}\right) = \min\left(1, \frac{p(y \mid \theta_{cand})\, p(\theta_{cand})}{p(y \mid \theta^{(t)})\, p(\theta^{(t)})}\right). \tag{5.7}$$

If the $p(\theta_{cand} \mid y)$ is larger than the $p\left(\theta^{(t)} \mid y\right)$, θ_{cand} is accepted since its acceptance probability equals 1. If it is lower than 1, a uniform random variable $u^{(t)}$ is generated, and the following rule is used for setting the next value of $\theta^{(t+1)}$:

$$\theta^{(t+1)} = \begin{cases} \theta_{cand} & \text{if } p^{(t)} \leq u^{(t)}, \\ \theta^{(t)} & \text{if } p^{(t)} > u^{(t)}. \end{cases} \tag{5.8}$$

By using that rule, the algorithm draws values more frequently where posterior density is higher, and randomness in Equation 5.8 allows values with lower posterior densities.

5.1.2 Hamiltonian Monte Carlo

Although Metropolis sampling provides a practical solution to find posterior densities, it is not an efficient algorithm. For a more efficient exploration, various

extensions of Metropolis sampling are offered. One of the most popular extensions of the Metropolis algorithm is called Hamiltonian Monte Carlo (HMC). The name of the algorithm refers to Hamiltonian motion equations. HMC uses a physics analogy to explore the parameter space more efficiently. Avoiding the random walk enables a more efficient exploration, especially when the parameter space is too complex or model parameters are correlated [22].

Hamiltonian motion equations determine the rate of change at momentum p and position q by taking the partial derivative of Hamiltonian function $H(q,p)$ [22]. Thus given time, changes in position q and momentum p can be calculated by solving the equations presented at (5.9). In the non-psychical applications, position parameter q is the vector of interested parameters; in the case of Bayesian sampling, it denotes the d-dimensional vector of model parameters. There is no actual meaning for momentum parameter p in the statistical application of HMC; instead, it is an auxiliary variable that provides more efficient position changing in the model parameter space.

Hamiltonian motion equations can be written as

$$\begin{aligned} \frac{dq}{dt} &= \frac{\partial H}{\partial p}, \\ \frac{dp}{dt} &= -\frac{\partial H}{\partial q}, \end{aligned} \tag{5.9}$$

and the Hamiltonian function can be defined as

$$H(q,p) = U(q) + K(p), \tag{5.10}$$

where $U(q)$ denotes the potential energy, and $K(p)$ represents the kinetic energy. For performing Bayesian sampling using Hamiltonian dynamics, posterior probabilities must be related to the potential energy function. On the other hand, the kinetic energy function uses a classical formula, where p as the momentum values and diagonal matrix M, with $m_1, \ldots, m_d$ on the diagonal, as the mass. Then

$$\begin{aligned} U(q) &= -\log\left[P(q \mid y)\right], \\ K(p) &= p^T M^{-1} p/2. \end{aligned} \tag{5.11}$$

In the sampling phase, candidates for parameters p and q can be selected using the Hamiltonian motion equations presented in (5.9). To be able to implement this, Equations 5.9 must be approximated using some discretization methods [22]. The leapfrog method is an efficient approximation method that uses a half-step($\epsilon/2$) for updating momentum p and takes a full step(ϵ) for updating position q, and again takes a half-step for updating momentum p:

$$\begin{aligned} p_i\left(t+\epsilon/2\right) &= p_i\left(t\right) - \left(\epsilon/2\right)\frac{\partial U}{\partial q_i}\left(q\left(t\right)\right), \\ q_i\left(t+\epsilon\right) &= q_i\left(t\right) + \epsilon\frac{p_i\left(t+\epsilon/2\right)}{m_i}, \\ p_i\left(t+\epsilon\right) &= p_i\left(t+\epsilon/2\right) - \left(\epsilon/2\right)\frac{\partial U}{\partial q_i}\left(q\left(t+\epsilon\right)\right). \end{aligned} \tag{5.12}$$

As the last step of this construction, the Hamiltonian function can be used as an energy function that appears in the canonical probability distribution in statistical mechanics. Canonical probability distribution of a system that has energy function E and states x can be defined as [22]

$$P\left(x\right) = \frac{1}{Z}\exp\left(\frac{-E\left(x\right)}{T}\right). \tag{5.13}$$

By setting $T = 1$, considering Z as some constant, and using $H\left(q,p\right)$ as the energy function joint probability of q and p has the density function

$$\begin{aligned} P\left(q,p\right) &= \frac{1}{Z}\exp\left(\frac{-H\left(q,p\right)}{1}\right) \\ &= \frac{1}{Z}\exp\left[-U\left(q\right)\right]\exp\left[-K\left(q\right)\right] \\ &= P\left(q \mid y\right)\frac{1}{Z}\exp\left(\frac{p^T M^{-1} p}{2}\right). \end{aligned} \tag{5.14}$$

The resultant joint probability has a quite nice form. It can be seen that auxiliary variable p is independent of model parameters q and $p \sim N\left(0, M\right)$. Thus, an efficient sampling of posterior density can be achieved by iteratively applying following steps [23]. To relate this procedure to previous discussions, let us use θ instead of q and ϕ instead of p.

1. Update ϕ by drawing a sample from $N(0, M)$.

2. Set $\theta^{(t)}$ to θ and $\phi^{(t)}$ to ϕ.

3. Update θ and ϕ L times by using the Leapfrog method that is presented in 5.12.

   Repeat L times:

   (a) $\phi = \phi + \frac{\epsilon}{2}\frac{d \log p(\theta|y)}{d\theta}$

   (b) $\theta = \theta + \epsilon M^{-1}\phi$

   (c) $\phi = \phi + \frac{\epsilon}{2}\frac{d \log p(\theta|y)}{d\theta}$

4. Set θ^{cand} to θ and ϕ^{cand} to ϕ.

5. Calculate the acceptance probability

   $$\alpha = \frac{p\big(\theta^{cand}|y\big)p\big(\phi^{cand}\big)}{p\big(\theta^{(t)}|y\big)p\big(\phi^{(t)}\big)}.$$

6. Set

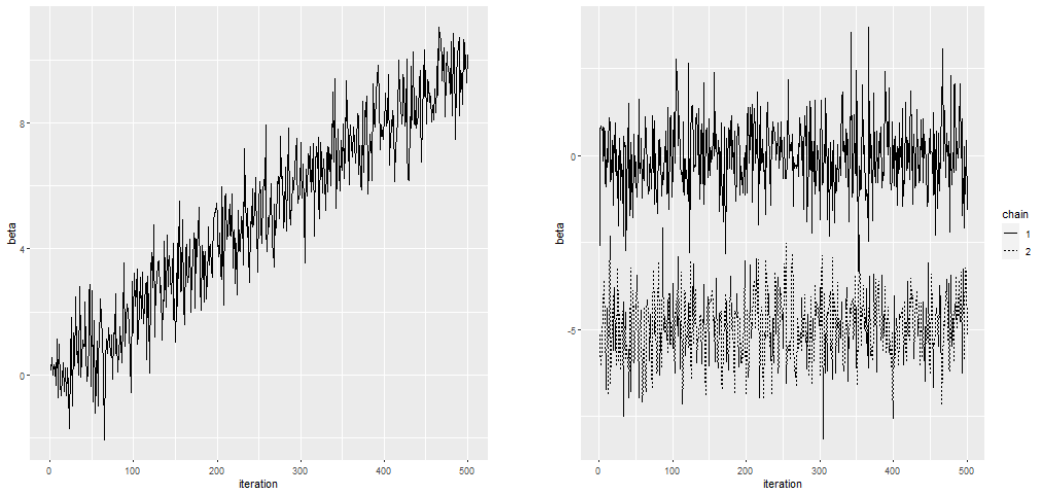
   $$\theta^{(t+1)} = \begin{cases} \theta_{cand} & \text{with probability } \alpha, \\ \theta^{(t)} & \text{otherwise.} \end{cases}$$

The effect of the momentum variable can be understood by considering scenarios according to position in parameter space. Let us assume we are taking steps to a flat area in parameter space. Then the $\frac{d \log p(\theta|y)}{d\theta}$ will equal zero, and we will change position at a constant speed. In another scenario, we are taking steps to a low-density area, and then the $\frac{d \log p(\theta|y)}{d\theta}$ will be negative. Thus, our speed will decrease until we make a return to the opposite direction, which is the high density area. After the direction of the algorithm points higher density area, the $\frac{d \log p(\theta|y)}{d\theta}$ will be positive, and momentum will increase until the direction points to a lower density area. So, the algorithm tends to explore denser areas, and the quality of drawn samples will increase [23].

5.2 Diagnostics and Comparison Methods

5.2.1 Chain Diagnostics

Two challenges arise in iterative sampling methods. Firstly, sampling chains sometimes may not have converged yet. In the early iterations of sampling, chains may have not reached to support of stationary distribution. So, a prematurely stopped sampling chain provides a posterior distribution that is not truly representative. In some cases, separately monitored chains seem to achieve convergence, but this can be a deceptive observation if chains do not converge to the same distribution. Examples of both convergence problems are demonstrated in Figure 5.1.



(a) Single chain (b) Multiple chains

Figure 5.1: Illustration of convergent problems

Although convergence problems can be related to insufficient iteration numbers, they can also indicate degeneracies and misspecifications in the model. When the number of iterations is believed to be sufficient, not converged or wrongly converged chains are a sign of strange geometries in parameters space. These intricate geometries can be caused by misspecified models or wrong implementations. In that case, debugging the implementation or revising the model

can be a solution. On the other hand, a theoretically reasonable model can have pathological degeneracies such as funnel degeneracies [24] or identifiability issues [25]. Both problems can create extremely narrow parameter spaces, and chains might be stuck in a local area or can not explore the parameter space in a finite time.

The second problem may arise when the draws of chains are correlated. Although an autocorrelated chain is not a serious issue, highly correlated samples indicate inefficient samplings [23]. As in the convergence problem, highly correlated samples might indicate degeneracies or misspecification.

5.2.1.1 $\hat{R}$ values

Monitoring sampling chains is a useful visualization method that examines whether chains are mixed and stationarity is achieved. However, visual diagnostics can be subjective in terms of understanding the mixing and should be supported by some metrics. For assessing the mixing of sampling chains, a metric $\hat{R}$ can be defined using between-sequence variance B and within-sequence variance W [23]. For achieving stationarity, iterations need to be reached to some point. Generally, 1000 iterations are enough to achieve stationarity for a nicely constructed model. Iterations up to this point are called warm-up iterations, and they are discarded when statistics related to posterior density are calculated. Let us assume that four independent chains were used for sampling with 2000 iterations. After discarding warm-up iterations, the remaining samples are divided into two parts. Then we obtain $m = 8$ chains with length $n = 250$. If we denote each sample as $\theta_{ij}\,(i = 1, \ldots, n; j = 1, \ldots, m)$, then we can calculate

$$\begin{aligned} B &= \frac{n}{m}\sum_{j=1}^{m}\left(\bar{\theta}_{.j} - \bar{\theta}_{..}\right), \text{ where} \quad \bar{\theta}_{.j} = \frac{1}{n}\sum_{i=1}^{n}\bar{\theta}_{ij},\ \bar{\theta}_{..} = \frac{1}{m}\sum_{j=1}^{m}\bar{\theta}_{.j}, \\ W &= \frac{1}{m}\sum_{j=1}^{m}s_j^2, \text{ where} \quad s^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(\bar{\theta}_{ij} - \bar{\theta}_{.j}\right). \end{aligned} \tag{5.15}$$

Then $\widehat{var}^{+}(\theta \mid y)$ can be estimated by taking the weighted average of between-sequence variance B and within-sequence variance W as in

$$\widehat{var}^{+}(\theta \mid y) = \frac{n-1}{n}W + \frac{1}{n}B. \tag{5.16}$$

For obtaining mixed chains, B should be close enough to W. For assessing this, the related metric can be defined as

$$\widehat{R} = \sqrt{\frac{\widehat{var}^{+}(\theta \mid y)}{W}}. \tag{5.17}$$

As B gets close to the W equation, $\widehat{R}$ will be closer to 1. $\widehat{R}$ values smaller than 1.1 are acceptable, and larger than 1.1 indicates chains are not mixed, and more samples may be needed.

5.2.1.2 Effective number of simulation draws

After ensuring mixing, the effective number of independent draws can be calculated. If samples are independently distributed, there will be mn independent draws. However, it's mostly not true in practice, and samples have autocorrelation [23]. Then the metric effective sample size can be defined as

$$n_{eff} = \frac{mn}{1 + 2\sum_{t=1}^{\infty}\rho_t}. \tag{5.18}$$

Here ρ_t is the autocorrelation of θ a lag t and can be calculated using

$$\widehat{\rho} = 1 - \frac{V_t}{2\widehat{var}^{+}}, \text{ where } \quad V_t = \frac{1}{m(n-t)}\sum_{j=1}^{m}\sum_{i=t+1}^{n}\left(\bar{\theta}_{i,j} - \bar{\theta}_{i-t,j}\right)^2. \tag{5.19}$$

As an arbitrary but pratical choice, chains with $\frac{n_{eff}}{mn} \geq 0.1$ can be considered as efficient sampling chains [26]. On the other hand, chains with an effective sample ratio smaller than 0.1 indicate a problematic sampling process.

5.2.2 Posterior Predictive Checks

Posterior predictive checking provides both visual and statistical diagnostics to assess the fitness of a Bayesian model. The idea is to produce new response variables y^{rep} using the posterior distribution of model parameters and compare them

with observed data y. Formally replications are generated from the conditional density of y^{rep} given y, which can be defined as

$$p\left(y^{rep} \mid y\right) = \int p\left(y^{rep} \mid \theta\right) p\left(\theta \mid y\right) d\theta. \tag{5.20}$$

In practice, posterior predictive distribution $p\left(y^{rep} \mid y\right)$ can be obtained by generating response variables using the drawn parameters in each iteration of the sampling phase. Then a p-value can be calculated using a test statistic $T\left(y\right)$. This test statistic can be a maximum, minimum, mean, median, and zero portion of the distribution.

$$p - \text{value} = Pr\left[T\left(y^{rep}\right) \geq T\left(y\right)\right] \tag{5.21}$$

For calculating the probability in Equation 5.21, the selected statistic T is applied to each sampled distribution of y. Then p-value can be calculated by looking at the portion y^{rep}'s that is $T\left(y^{rep}\right) \geq T\left(y\right)$. The visualization of this is presented in Figure 5.2.

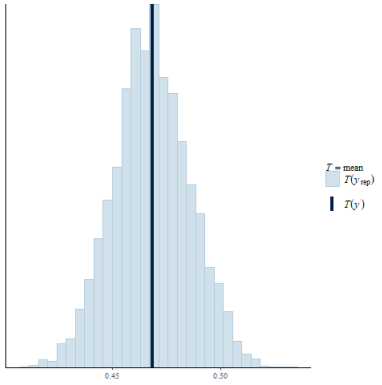


Figure 5.2: Posterior predictive check on mean

Posterior predictive distribution can also be used to compare the frequencies of actual and replicated responses. This comparison can be visualized with posterior retrodictive plots. These plots draw the histogram of each response in replicated data and show the frequencies of actual responses in the same plot. For example,

it can be understood whether the zero counts in the prediction match the exact zero count. An example is presented in Figure 5.3

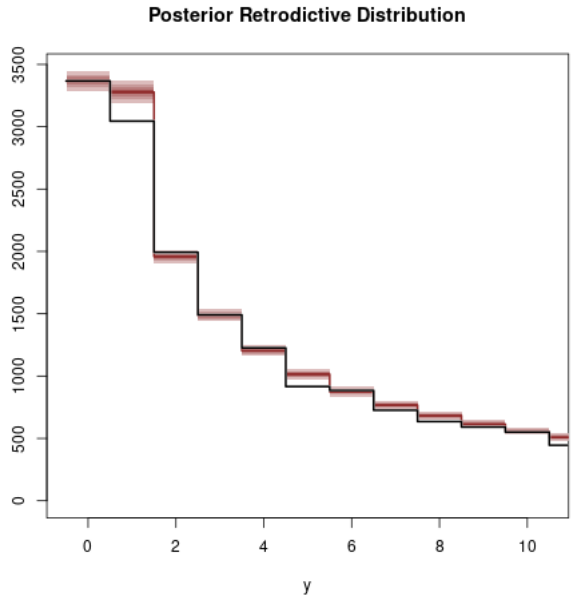


Figure 5.3: Posterior retrodictive plot

Here black lines indicate the number of counts in the data for the values shown on the x-axis. Red areas represent the distribution of frequencies, and lighter areas speak for less dense regions in the predictive distribution.

5.2.3 Loo and WAIC

Loo stands for the leave-one-out cross-validation, and it is the special case of K-fold cross-validation that K equals to sample size of the data. So, calculation of it is quite computational costly since each model needs to be fitted K times. The computation time of such an application is not feasible when the data set size is large. On the other hand, approximation of Loo is possible using importance sampling, and Vehtari, Gelman, and Gabry [27] proposed a more robust method that uses Pareto distribution in the importance sampling. To be able to assess the models, a measure called loo estimate of expected log pointwise predictive

density ($elpd_{loo}$) is defined as

$$elpd_{loo} = \sum_{i=1}^{n} \log p\left(y_i \mid y_{-i}\right), \tag{5.22}$$

where

$$p\left(y_i \mid y_{-i}\right) = \int p\left(y_i \mid \theta\right) p\left(\theta \mid y_{-i}\right) d\theta. \tag{5.23}$$

The predictive distribution of observation y_i in Equation 5.23 can be estimated using ordinary importance sampling. Let us denote the importance weights of observation i from the sample s

$$r_i^s = \frac{1}{p\left(y_i \mid \theta^s\right)} \propto \frac{p\left(\theta^s \mid y_{-i}\right)}{p\left(\theta^s \mid y\right)}. \tag{5.24}$$

Then the approximation of $p\left(y_i \mid y_{-i}\right)$ can be defined as

$$p\left(y_i \mid y_{-i}\right) \approx \frac{\sum_{s=1}^{S} r_i^s p\left(y_i \mid \theta^s\right)}{\sum_{s=1}^{S} r_i^s}. \tag{5.25}$$

However, this method is unstable, and sometimes the variance of the distribution of importance ratios can be infinite. Thus the proposed methods fit a Pareto distribution to some portion of the largest values of importance ratios and smooth the upper tail of the distribution by replacing these largest values with the expected order statistics of the fitted Pareto distribution. Furthermore, the estimated shape parameter of the Pareto distribution k can be used to assess reliability. The results indicate that sampling with k larger than 0.7 is problematic. If the ratio of problematic transactions is too high, then the practitioner should use more computationally costly methods like K-fold cross-validation [27]. Finally, the estimated loo can be defined as

$$\widehat{loo} = -2 \cdot elpd_{loo}. \tag{5.26}$$

The widely applicable AIC or Watanabe AIC (WAIC) is another metric for making a Bayesian model comparison, and asymptotically it is similar to the loo.

$$elpd_{waic} = \sum_{i=1}^{n} \log\left(y_i \mid y\right) - \sum_{i=1}^{n} var_{post}\left[\log p\left(y_i \mid \theta\right)\right], \tag{5.27}$$

Here posterior variance can be calculated using the sample variance formula. Similar to the loo, estimated WAIC can be approximated using the formula

$$\widehat{WAIC} = -2 \cdot elpd_{waic}. \tag{5.28}$$

In both metrics, smaller values indicate a better fit since they provide higher predictive accuracy on unseen observations.

Chapter 6

Hierarchical Bayesian Models for Estimating Number of Installed Refrigerators

In this chapter, various types of hierarchical Bayesian models are introduced, and diagnostics are performed. Models have a similar hierarchical structure, meaning that the same covariates and grouping factors are used. For forming the grouping factor, the refrigerator and compressor models are combined. The refrigerator's installation and production months are used as covariates, and the age covariate is discarded since it does not improve the models' prediction accuracy. The difference between models is the likelihood function of the response variable, and this difference makes some models more suitable due to overdispersion and zero counts in the data.

The Bayesian estimations of models are performed using Stan, a probabilistic programming language. Stan uses a modified version of HMC that automatically tunes some hyperparameters such as step size and the number of steps in each iteration. Each estimation is performed with four parallel chains and 2000 iterations. The iteration numbers seem inadequate for those familiar with other Bayesian sampling methods, such as Gibbs sampling. However, the quality of

these samples is improved thanks to the efficient exploration [28]. Unfortunately, smaller iteration numbers can't provide a shorter run time since each iteration requires exhausting calculations.

6.1 Models

We used five different count data models: Poisson, binomial, negative binomial, beta-binomial, and hurdle. The general structure of the models is presented by Directed Acyclic Graph (DAG) in Figure 6.1.

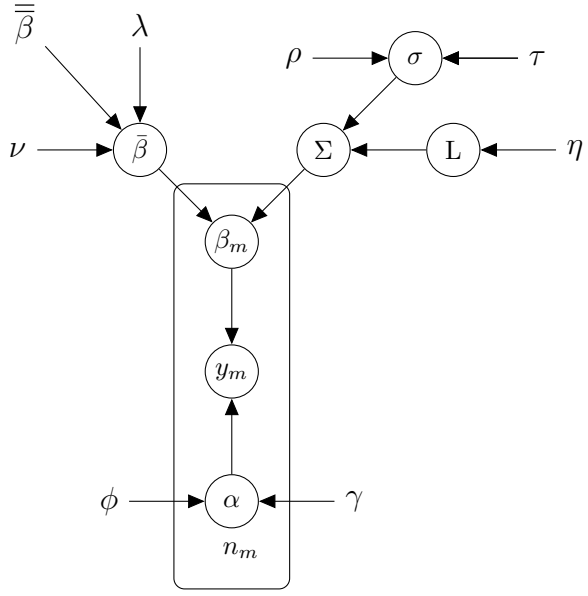


Figure 6.1: DAG for hierarchical models

6.1.1 Poisson Model

In this model, we used the Poisson likelihood function to model the number of installed refrigerators. The structure of the model can be defined as follows.

$$
\begin{aligned}
y_m &\sim \text{Pois}\left(\mu_m\right), \\
\mu_m &= \exp\left(X_m\beta_m + \log\left(n_m\right)\right), \\
\beta_m &\sim \mathcal{N}(\bar{\beta}, \Sigma), \\
\bar{\beta} &\sim t_3\left(0, 10\right), \\
\Sigma &= \sigma L \sigma^T, \\
L &\sim \text{LKJCorr}\left(2\right), \\
\sigma &\sim C^+\left(0, 3\right).
\end{aligned}
\tag{6.1}
$$

Here m denotes the grouping factor; namely, the refrigerator model. The covariates' effect $\bar{\beta}$ is used as the mean of group-level effects β. Each covariate effect $\bar{\beta}$ is independently and identically distributed with Student's t-distribution with center at zero and scale 10. Here prior for $\bar{\beta}$ can be considered as weakly informative since it has enough large standard deviation. Student's t-distribution with the degree of freedom 3 has heavier tails than the normal distribution; thus, it allows larger values and might be preferable. On the other hand, using a smaller degree of freedom can cause longer run times or strange geometries that can't be easily explored. The group-level effects β_m has multivariate normal distribution with covariance matrix Σ generated with group-level standard deviation σ and correlation matrix L. Each σ has half-Cauchy distribution with mean zero and a scale parameter equal to 3. It is a better alternative to inverse-gamma distribution since posterior distribution can be too sensitive to the choice of parameters of inverse-gamma distribution [29]. The correlation matrix L has an LKJ correlation distribution with a shape parameter equal to 2. For larger shape parameter values, the correlation matrix is more like the identity matrix [30]. This distribution is widespread among Stan users since its alternative inverse-Wishart distribution creates unstable calculations in Stan. Lastly, the number of refrigerators ready for installation is denoted with n_m and used as an offset.

6.1.2 Negative Binomial Model

In this model, we used the negative binomial likelihood function to model the number of installed refrigerators. The structure of the model can be defined as in Equation 6.2.

Here we have the same structure as in the Poisson model. Additionally, we defined the overdispersion parameter α, which relaxes the equal mean and variance assumption of the Poisson model. The overdispersion parameter α has a gamma distribution with both scale and shape parameters equal to 0.01. This is a weakly informative prior since the distribution is centered at 1 with a long right tail. So, values smaller than 1 are more likely, but larger values are also allowed.

$$
\begin{aligned}
y_m &\sim \mathrm{NB}\left(\mu_m, \alpha\right), \\
\mu_m &= \exp\left(X_m \beta_m + \log\left(n_m\right)\right), \\
\beta_m &\sim \mathcal{N}(\bar{\beta}, \Sigma), \\
\bar{\beta} &\sim t_3\left(0, 10\right), \\
\Sigma &= \sigma L \sigma^T, \\
L &\sim \mathrm{LKJCorr}\left(2\right), \\
\sigma &\sim C^{+}\left(0, 3\right), \\
\alpha &\sim \mathrm{Gamma}\left(0.01, 0.01\right).
\end{aligned}
\tag{6.2}
$$

6.1.3 Binomial Model

In this model, we used the binomial likelihood function to model the number of installed refrigerators. The structure of the model can be defined as in Equation 6.3

$$
\begin{aligned}
y_m &\sim B\left(n_m, \mu_m\right), \\
\text{logit}\left(\mu_m\right) &= X_m \beta_m, \\
\beta_m &\sim \mathcal{N}(\bar{\beta}, \Sigma), \\
\bar{\beta} &\sim t_3\left(0, 10\right), \\
\Sigma &= \sigma L \sigma^T, \\
L &\sim \text{LKJCorr}\left(2\right), \\
\sigma &\sim C^+\left(0, 3\right).
\end{aligned}
\tag{6.3}
$$

6.1.4 Beta-Binomial Model

In this model, we used the beta-binomial likelihood function. The structure of the beta-binomial model can be defined as in Equation 6.4

$$
\begin{aligned}
y_m &\sim \text{BetaBin}\left(n_m, \mu_m, \alpha\right), \\
\text{logit}\left(\mu_m\right) &= X_m \beta_m, \\
\beta_m &\sim \mathcal{N}(\bar{\beta}, \Sigma), \\
\bar{\beta} &\sim t_3\left(0, 10\right), \\
\Sigma &= \sigma L \sigma^T, \\
L &\sim \text{LKJCorr}\left(2\right), \\
\sigma &\sim C^+\left(0, 3\right), \\
\alpha &\sim \text{Gamma}\left(0.01, 0.01\right).
\end{aligned}
\tag{6.4}
$$

Additional to the binomial distribution, we have an overdispersion parameter α analogous to the negative binomial distribution.

6.1.5 Negative Binomial Hurdle Model

$$
\begin{aligned}
y_m \mid y_m > 0 &\sim \mathrm{NB}\left(\mu_m, \alpha\right), \\
\mathbb{P}\left(y_m = 0\right) &= \pi_m, \\
\mathrm{logit}\left(\pi_m\right) &= X_m \gamma_m, \\
\log\left(\mu_m\right) &= X_m \beta_m + \log\left(n_m\right), \\
\log\left(\alpha\right) &= X\theta, \\
\gamma_m &\sim \mathcal{N}(\bar{\gamma}, \Sigma_\pi), \\
\beta_m &\sim \mathcal{N}(\bar{\beta}, \Sigma_\mu), \\
\bar{\gamma} &\sim t_3\left(0, 10\right), \\
\bar{\beta} &\sim t_3\left(0, 10\right), \\
\theta &\sim t_7\left(0, 10\right), \\
\Sigma_\pi &= \sigma_\pi L_\pi \sigma_\pi^T, \\
L_\pi &\sim \mathrm{LKJCorr}\left(2\right), \\
\sigma_\pi &\sim C^+\left(0, 3\right), \\
\Sigma_\mu &= \sigma_\mu L_\mu \sigma_\mu^T, \\
L_\mu &\sim \mathrm{LKJCorr}\left(2\right), \\
\sigma_\mu &\sim C^+\left(0, 3\right).
\end{aligned}
\tag{6.5}
$$

In this model, we use two different likelihood functions to express positive and zero counts separately. The first part of the model has a binomial likelihood function that decides whether the response will be equal to zero or not. The second part models the positive counts, and we used a truncated negative binomial likelihood function for this purpose. We select negative binomial distribution because we believe that data is overdispersed since there are many distinct values from the mean, even if zeros are not included. In the first version of this model probability of being zero did not have covariates, but we realized that this model tended to over-estimate zero counts since there are many small positive integers in the data. On the other hand, adding covariates to this probability might create overfitting issues since small numbers are likely to appear as zero in the unseen data. Nevertheless, we decided to add covariates for the binomial part

since approximated cross-validation scores suggested that. Also, we believe that dispersion in positive counts might depend on refrigerator type. So, we estimate inverse-dispersion parameter α using the refrigerator and compressor model as a covariate.

6.2 Diagnostics

This section presents the summary of proposed hierarchical models and the results of Bayesian diagnostic tools. Variables that appear in model summaries are described in Table 2.1.

6.2.1 Poisson Model Diagnostics

Listing 6.1 shows the Poisson model summary. The first block gives information about the structure of the model and the sampling. The second block presents the group-level effects, and the third shows the population-level effects. For the model parameters, means of their posterior distribution, estimated errors, 95% credible intervals, and $\hat{R}$ values are presented in Listing 6.1.

Population-level installation month effects of June (InstMon6), July (InstMon7), August (InstMon8), and September (InstMon9),on installation numbers are more significant than the effects of other months. On the other hand, population-level production month effects are not very strong; some are insignificant since their credible intervals include zero. On the other hand, population-level production month effects are not very strong, and most of them are insignificant since their credible intervals include zero. Group-level effects are strong enough to indicate the refrigerator and compressor model influence population-level effects.

Listing 6.1: Summary of Poisson model

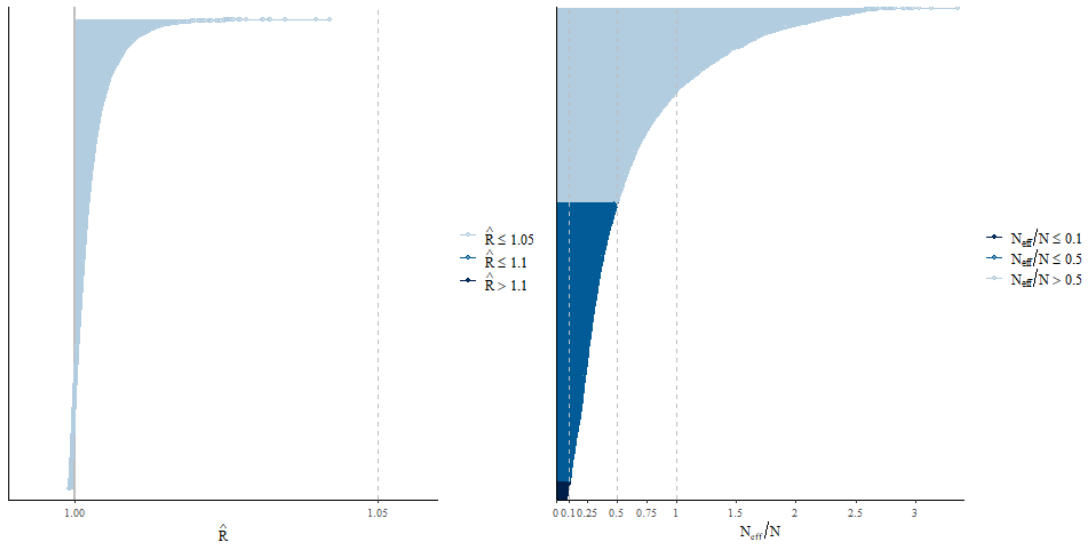
```
Family: poisson
Links: mu = log
Formula: Installed ~ 1 + InstMon + ProdMon
                   + offset(log(Produced))
                   + (1 + InstMon + ProdMon | Model)
Data: InstallationTrain (Number of observations: 15321)
Draws: 4 chains, each with iter = 1000; warmup = 0; thin = 1;
          total post-warmup draws = 4000

Group-Level Effects:
~Model (Number of levels: 141)
                Estimate Est.Error l-95% CI u-95% CI Rhat
sd(Intercept)       0.78      0.05       0.69       0.88 1.00
sd(InstMon2)        0.61      0.05       0.52       0.72 1.01
sd(InstMon3)        0.86      0.06       0.74       0.99 1.01
sd(InstMon4)        0.85      0.06       0.73       0.98 1.01
sd(InstMon5)        0.77      0.06       0.67       0.89 1.01
sd(InstMon6)        0.61      0.04       0.53       0.70 1.01
sd(InstMon7)        0.85      0.06       0.74       0.97 1.01
sd(InstMon8)        0.93      0.06       0.83       1.05 1.01
sd(InstMon9)        0.70      0.04       0.62       0.79 1.01
sd(InstMon10)       0.71      0.05       0.63       0.81 1.00
sd(InstMon11)       0.60      0.04       0.52       0.69 1.01
sd(InstMon12)       0.63      0.05       0.53       0.73 1.01
sd(ProdMon2)        0.55      0.05       0.45       0.66 1.00
sd(ProdMon3)        0.69      0.06       0.58       0.82 1.00
sd(ProdMon4)        0.47      0.04       0.38       0.56 1.00
sd(ProdMon5)        0.63      0.05       0.54       0.75 1.00
sd(ProdMon6)        0.52      0.04       0.44       0.61 1.00
sd(ProdMon7)        0.67      0.05       0.57       0.78 1.00
sd(ProdMon8)        0.65      0.05       0.56       0.76 1.00
sd(ProdMon9)        0.66      0.06       0.54       0.80 1.00
sd(ProdMon10)       0.60      0.05       0.51       0.72 1.00
```

sd(ProdMon11)	0.74	0.07	0.61	0.89	1.00
sd(ProdMon12)	0.67	0.07	0.56	0.81	1.00

Population–Level Effects:

	Estimate	Est.Error	l–95% CI	u–95% CI	Rhat
Intercept	−3.25	0.07	−3.39	−3.11	1.01
InstMon2	0.14	0.07	0.02	0.27	1.02
InstMon3	0.45	0.08	0.29	0.61	1.02
InstMon4	0.38	0.08	0.22	0.54	1.01
InstMon5	0.72	0.07	0.57	0.86	1.01
InstMon6	0.97	0.06	0.85	1.08	1.02
InstMon7	1.35	0.08	1.20	1.50	1.01
InstMon8	1.40	0.08	1.23	1.56	1.01
InstMon9	1.12	0.07	0.99	1.24	1.01
InstMon10	0.82	0.07	0.69	0.94	1.02
InstMon11	0.57	0.06	0.45	0.68	1.01
InstMon12	0.42	0.06	0.30	0.54	1.01
ProdMon2	−0.12	0.06	−0.25	−0.02	1.00
ProdMon3	−0.18	0.07	−0.33	−0.04	1.00
ProdMon4	−0.13	0.05	−0.23	−0.03	1.00
ProdMon5	−0.11	0.07	−0.25	0.02	1.01
ProdMon6	−0.08	0.06	−0.19	0.03	1.00
ProdMon7	−0.13	0.07	−0.27	0.01	1.00
ProdMon8	−0.12	0.07	−0.26	0.02	1.00
ProdMon9	−0.01	0.07	−0.15	0.14	1.00
ProdMon10	0.06	0.07	−0.08	0.20	1.00
ProdMon11	−0.12	0.08	−0.29	0.04	1.00
ProdMon12	0.07	0.08	−0.09	0.24	1.00

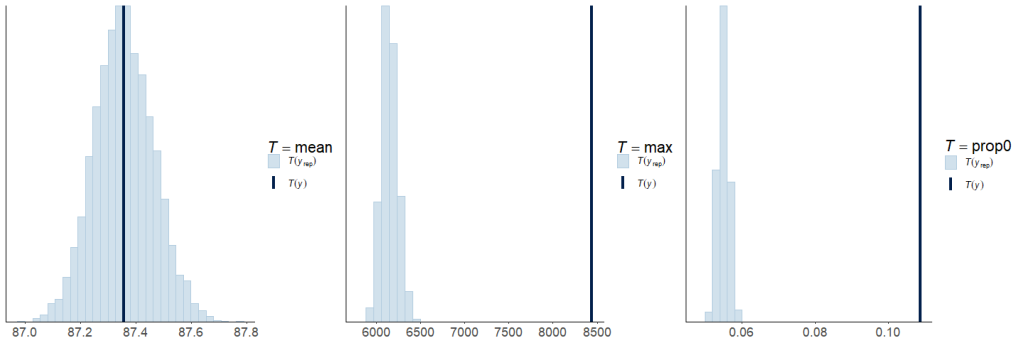
Chain diagnostics of the Poisson model are presented in Figure 6.2. $\widehat{R}$ values indicate that sampling chains mixed, meaning that they converged to the same distribution. On the other hand, there are many parameter samples with a $neff$ ratio lower than 0.1. This might indicate that the posterior distribution of some parameters is not reliable.



(a) $\widehat{R}$ values (b) Ratio of effective sample size

Figure 6.2: Chain diagnostics

The posterior predictive checks on the Poisson model are presented in Figure 6.3. We used the distribution's mean, max, and zero portions as $T(y)$. Since plots are obvious, we did not calculate $P\left[T\left(y^{rep}\right) \geq T(y)\right]$. The plot on the left shows the behavior of the mean and the posterior predictive distribution nicely centered around the mean of the actual data. On the other hand, the middle and right plots show that the model has a poor performance in terms of matching with maximum values and zero counts.

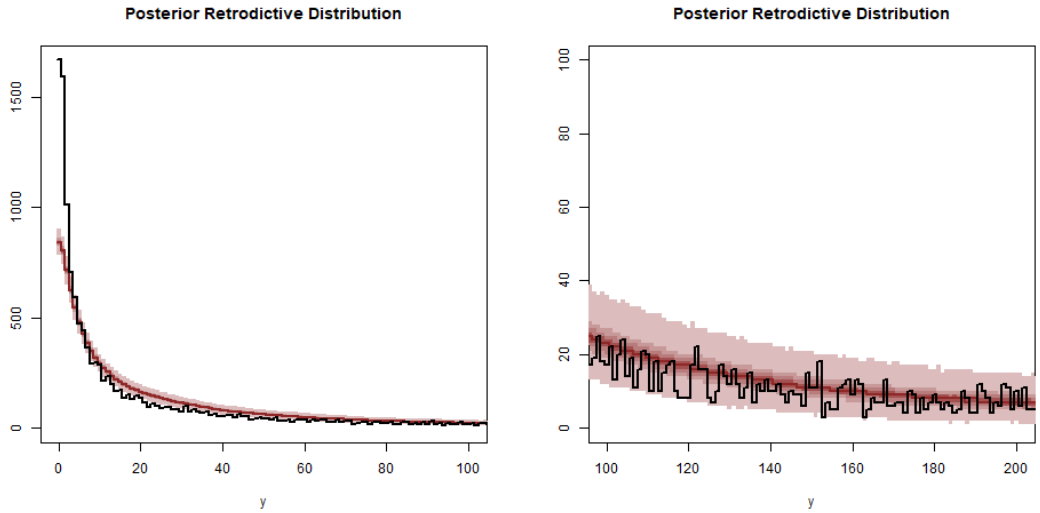


(a) Mean (b) Max (c) Zero portion

Figure 6.3: Posterior predictive check on mean, max, and zero portion

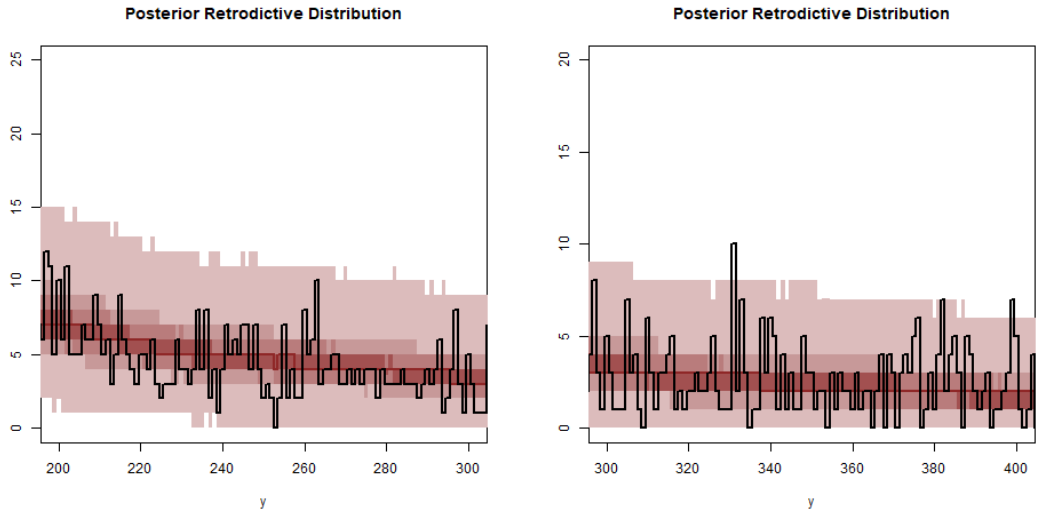
The posterior retrodictive check for different intervals is presented in Figure 6.4. In the figure, the lightest brown area represents 99%, the darkest brown area represents 20%, and the shades of brown between the lightest and darkest represent the 60%, 40% of the posterior retrodictive distribution. Since the scale of frequencies are quite different, we used four intervals to obtain better visualization. The reader should notice that there are y values larger than 400, but their frequencies are considerably low; thus, we only present the y values up to 400. The plot for the first interval indicates that the Poisson model under-estimates the frequencies of small numbers, and unfortunately, the difference is unreasonably high. After some value of y, the models start to over-estimate the counts until y reaches 150. For larger values model performs better, but still, there are some values out of 99% of posterior retrodictive distribution.

In Figure 6.5, we draw the 95% posterior predictive intervals according to the levels of installation month. The light blue points close to dark blue points indicate accurate predictions. It can be seen that there are many over- and under-estimations in the plot. Figure 6.6 shows the 95% posterior predictive intervals for each production month. Similarly, this plot indicates that predictions are not very accurate.



(a) Interval [0,100]

(b) Interval [100,200]



(c) Interval [200,300]

(d) Interval [300,400]

Figure 6.4: Posterior retrodictive check

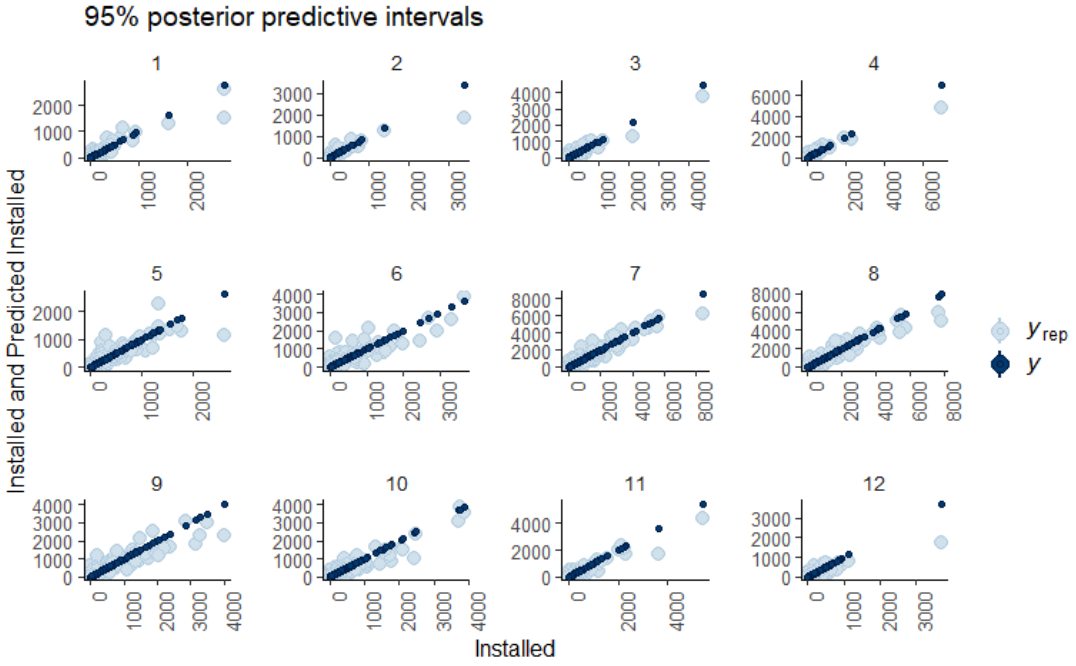


Figure 6.5: Installation month versus posterior predictive distribution

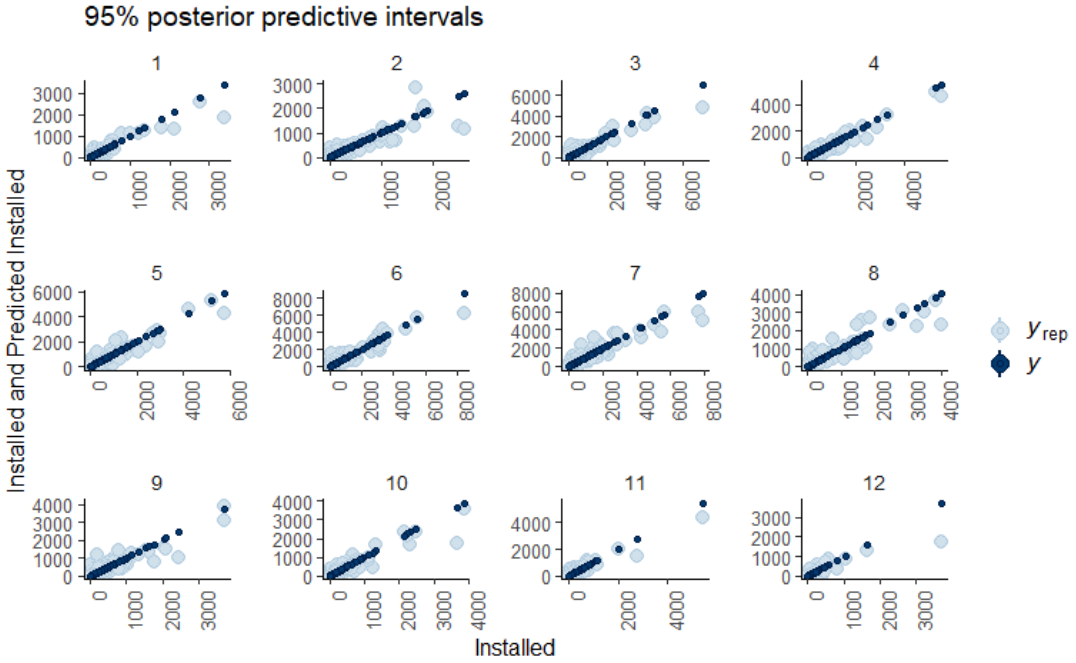


Figure 6.6: Production month versus posterior predictive distribution

6.2.2 Negative Binomial Model Diagnostics

Listing 6.2 shows the negative binomial model summary. Similar to the Poisson model, population-level installation month effects of June, July, August, and September are stronger than effects of other installation months. Although most of the population-level the production month effects are significant, their influence is not prominent. The group-level effects of the negative binomial model are weaker than the group-level effects of the Poisson model. The inverse-overdispersion parameter is denoted with shape in the Listing 6.2 and its estimation indicates that data is overdispersed.

Listing 6.2: Summary of negative binomial model

```
2  Family: negbinomial
3  Links: mu = log; shape = identity
4  Formula: Installed ~ 1 + InstMon + ProdMon
5                         + offset(log(Produced))
6                         + (1 + InstMon + ProdMon | Model)
7  Data: InstallationTrain (Number of observations: 15321)
8  Draws: 4 chains, each with iter = 1000; warmup = 0; thin = 1;
9           total post-warmup draws = 4000

11 Group-Level Effects:
12 ~Model (Number of levels: 141)
13                 Estimate Est.Error l-95% CI u-95% CI Rhat
14 sd(Intercept)       0.45      0.03      0.39      0.51 1.00
15 sd(InstMon2)        0.20      0.07      0.04      0.33 1.01
16 sd(InstMon3)        0.11      0.06      0.01      0.22 1.01
17 sd(InstMon4)        0.27      0.05      0.18      0.38 1.01
18 sd(InstMon5)        0.09      0.05      0.01      0.20 1.00
19 sd(InstMon6)        0.11      0.05      0.01      0.21 1.00
20 sd(InstMon7)        0.21      0.04      0.15      0.28 1.00
21 sd(InstMon8)        0.39      0.04      0.31      0.46 1.00
22 sd(InstMon9)        0.26      0.04      0.19      0.33 1.00
23 sd(InstMon10)       0.22      0.04      0.15      0.31 1.00
```

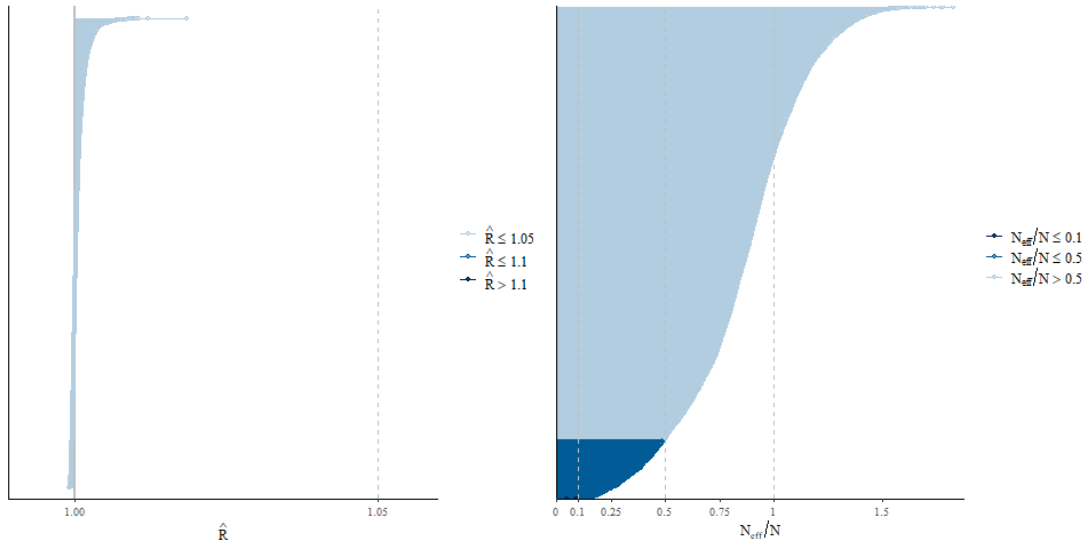
sd(InstMon11)	0.18	0.05	0.08	0.28	1.00
sd(InstMon12)	0.30	0.05	0.21	0.39	1.00
sd(ProdMon2)	0.21	0.06	0.09	0.33	1.00
sd(ProdMon3)	0.19	0.06	0.05	0.31	1.01
sd(ProdMon4)	0.11	0.06	0.01	0.22	1.01
sd(ProdMon5)	0.37	0.05	0.27	0.48	1.00
sd(ProdMon6)	0.08	0.05	0.00	0.19	1.00
sd(ProdMon7)	0.14	0.06	0.02	0.25	1.01
sd(ProdMon8)	0.10	0.06	0.01	0.22	1.00
sd(ProdMon9)	0.26	0.05	0.17	0.37	1.00
sd(ProdMon10)	0.37	0.05	0.27	0.48	1.00
sd(ProdMon11)	0.22	0.06	0.11	0.34	1.00
sd(ProdMon12)	0.34	0.06	0.22	0.47	1.00

Population-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat
Intercept	−3.24	0.06	−3.35	−3.12	1.00
InstMon2	0.16	0.05	0.06	0.26	1.00
InstMon3	0.47	0.05	0.38	0.56	1.00
InstMon4	0.45	0.05	0.35	0.56	1.00
InstMon5	0.78	0.04	0.69	0.86	1.00
InstMon6	1.04	0.04	0.96	1.13	1.00
InstMon7	1.38	0.05	1.29	1.46	1.01
InstMon8	1.33	0.05	1.22	1.44	1.00
InstMon9	0.99	0.05	0.90	1.09	1.00
InstMon10	0.68	0.05	0.59	0.77	1.00
InstMon11	0.37	0.05	0.28	0.46	1.00
InstMon12	0.28	0.05	0.18	0.38	1.00
ProdMon2	−0.12	0.04	−0.20	−0.03	1.00
ProdMon3	−0.16	0.04	−0.24	−0.08	1.00
ProdMon4	−0.13	0.04	−0.21	−0.05	1.00
ProdMon5	−0.16	0.05	−0.26	−0.05	1.00
ProdMon6	−0.07	0.04	−0.15	0.01	1.00
ProdMon7	−0.08	0.04	−0.15	0.00	1.00
ProdMon8	0.03	0.04	−0.05	0.12	1.00

```
ProdMon9        0.11      0.05      0.01      0.21 1.00
ProdMon10       0.18      0.06      0.07      0.30 1.00
ProdMon11       0.08      0.05     -0.02      0.18 1.00
ProdMon12       0.14      0.06      0.02      0.26 1.00

Family Specific Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat
shape     1.34      0.02      1.30      1.38 1.00
```

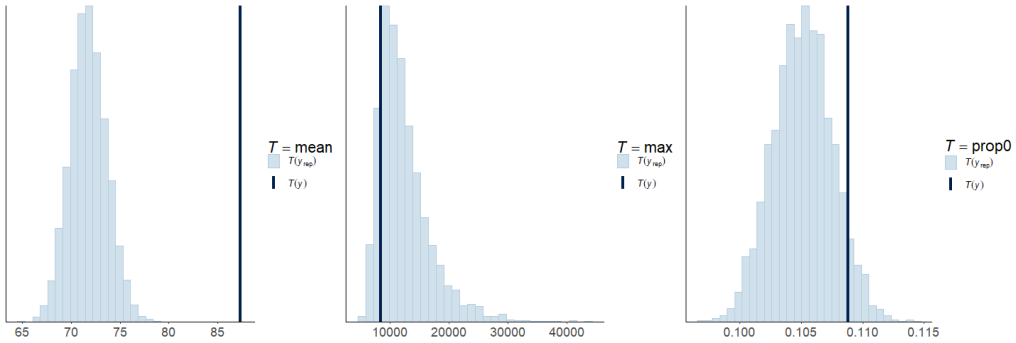
The chain diagnostic plots in Figure 6.7 show that the negative binomial model has a better mixing and effective sampling quality than the Poisson model. Since there are no samples with n_{eff} ratio lower than 0.1, exploration of the parameter space should be reliable.



(a) $\widehat{R}$ values (b) Ratio of effective sample size

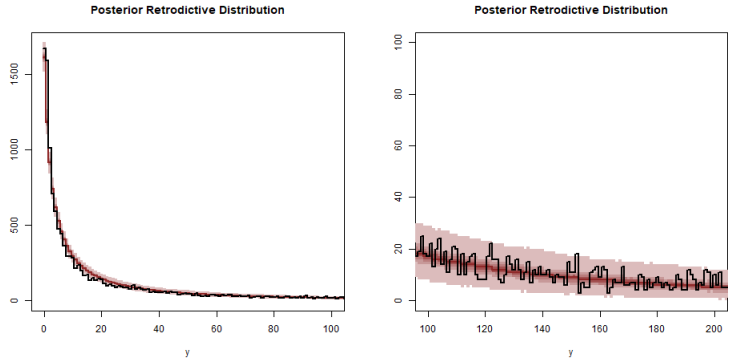
Figure 6.7: Chain diagnostics

Contrary to posterior predictive plots of the Poisson model in Figure 6.3, Figure 6.8 shows that the negative binomial model can match with larger values, and the zero portion is much closer to the actual value. Nonetheless, frequencies of means are distorted and under-estimate the true value.

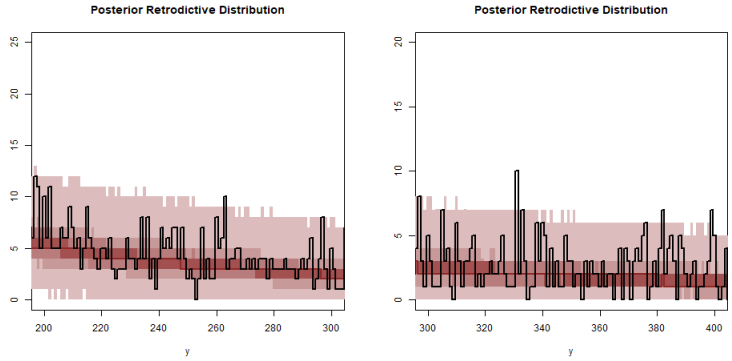


(a) Mean (b) Max (c) Zero Portion

Figure 6.8: Posterior predictive check on mean, max, and zero portion



(a) Interval [0,100] (b) Interval [100,200]



(c) Interval [200,300] (d) Interval [300,400]

Figure 6.9: Posterior retrodictive check

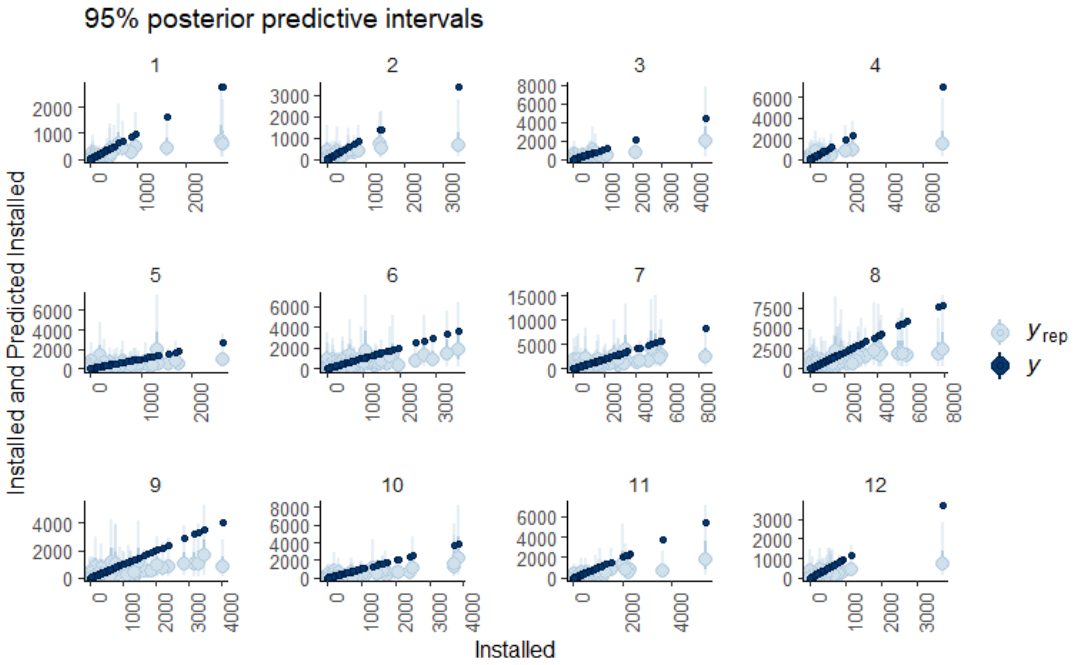


Figure 6.10: Posterior predictive distribution for each installation month

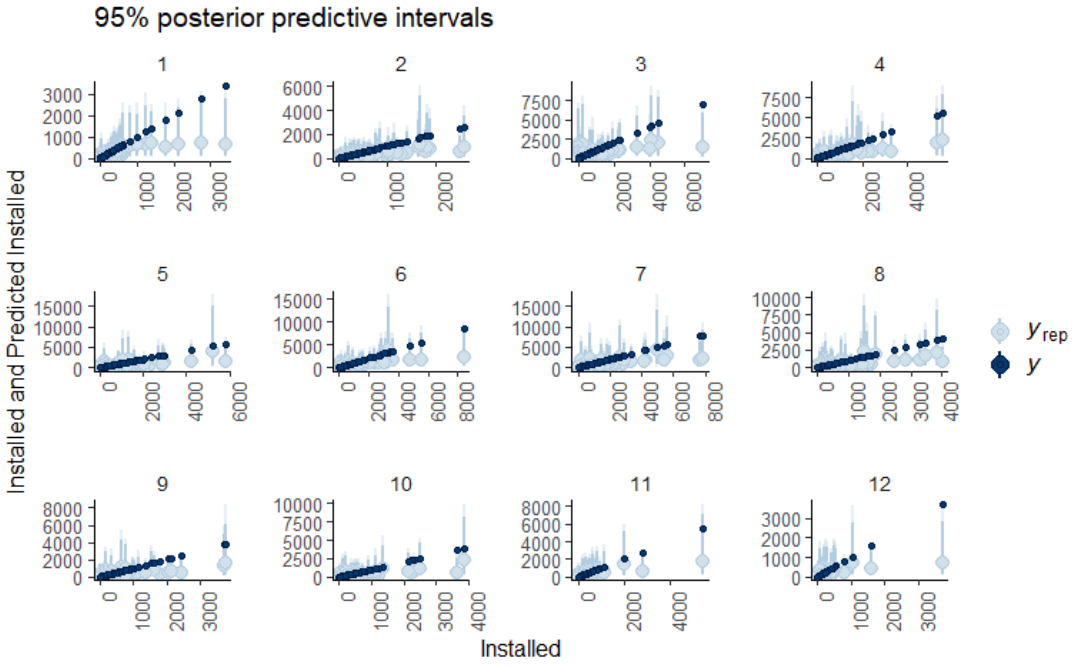


Figure 6.11: Posterior predictive distribution for each production month

By looking the Figure 6.9a, one can deduce that the negative binomial model has a similar kind of behavior to the Poisson model for the y values between 0 and 100. Of course, the amount of over- and under-estimation of counts is more dramatic for the Poisson model. One may notice that estimations of the counts for y values between 100 and 200 are also improved. However, for larger values of y, the performance of the negative binomial model is the same or slightly worse than the Poisson model.

Figures 6.10 and 6.11 show that the negative binomial model under-estimates larger values. Moreover, the 95% posterior intervals are extremely large, and predictions are unstable.

6.2.3 Binomial Model Diagnostics

The summary of the binomial model is presented in Listings 6.3, and chain diagnostics are shown in Figure 6.12. Both model summary and chain diagnostics plots of the binomial model are similar to the summary and chain diagnostics plots of the Poisson model. One can make similar interpretations.

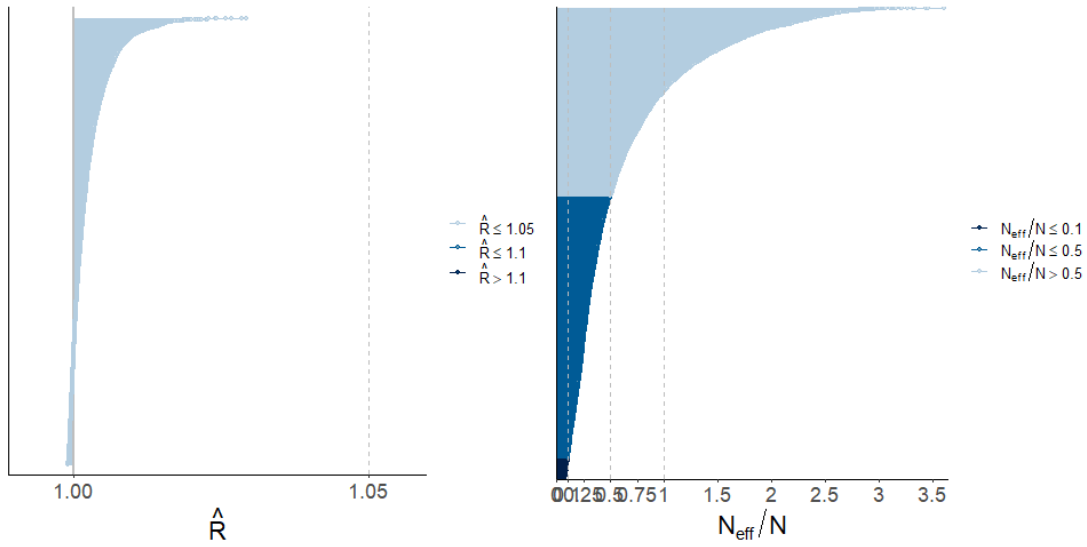
Listing 6.3: Summary of binomial model

```
2  Family: binomial
3  Links: mu = logit
4  Formula: Installed | trials(Produced) ~ 1 + InstMon + ProdMon
5                     + (1 + InstMon + ProdMon | Model)
6  Data: InstallationTrain (Number of observations: 15321)
7  Draws: 4 chains, each with iter = 1000; warmup = 0; thin = 1;
8           total post-warmup draws = 4000

10 Group-Level Effects:
11 ~Model (Number of levels: 141)
12                  Estimate Est.Error l-95% CI u-95% CI
13 sd(Intercept)        0.85      0.06     0.74     0.97
14 sd(InstMon2)         0.65      0.05     0.56     0.76
```

```
sd(InstMon3)          0.95      0.07      0.82      1.10
sd(InstMon4)          0.92      0.07      0.79      1.05
sd(InstMon5)          0.86      0.06      0.75      0.99
sd(InstMon6)          0.69      0.05      0.60      0.79
sd(InstMon7)          0.97      0.07      0.85      1.11
sd(InstMon8)          1.11      0.07      0.99      1.25
sd(InstMon9)          0.79      0.05      0.70      0.90
sd(InstMon10)         0.80      0.06      0.70      0.91
sd(InstMon11)         0.66      0.05      0.57      0.76
sd(InstMon12)         0.70      0.06      0.59      0.82
sd(ProdMon2)          0.63      0.06      0.52      0.75
sd(ProdMon3)          0.78      0.07      0.66      0.92
sd(ProdMon4)          0.54      0.05      0.45      0.65
sd(ProdMon5)          0.75      0.06      0.64      0.89
sd(ProdMon6)          0.64      0.05      0.53      0.75
sd(ProdMon7)          0.79      0.06      0.67      0.93
sd(ProdMon8)          0.80      0.06      0.69      0.93
sd(ProdMon9)          0.78      0.07      0.65      0.94
sd(ProdMon10)         0.68      0.06      0.58      0.81
sd(ProdMon11)         0.82      0.08      0.67      0.99
sd(ProdMon12)         0.77      0.07      0.64      0.93

Population-Level Effects:
             Estimate Est.Error l-95% CI u-95% CI Rhat
Intercept       -3.20      0.08     -3.36     -3.05 1.02
InstMon2         0.15      0.07      0.01      0.28 1.01
InstMon3         0.49      0.09      0.32      0.66 1.01
InstMon4         0.41      0.09      0.24      0.58 1.01
InstMon5         0.79      0.08      0.63      0.95 1.03
InstMon6         1.06      0.07      0.93      1.20 1.01
InstMon7         1.53      0.09      1.35      1.70 1.01
InstMon8         1.58      0.10      1.38      1.78 1.02
InstMon9         1.22      0.07      1.09      1.37 1.01
InstMon10        0.88      0.08      0.74      1.03 1.01
InstMon11        0.60      0.06      0.48      0.73 1.01
```

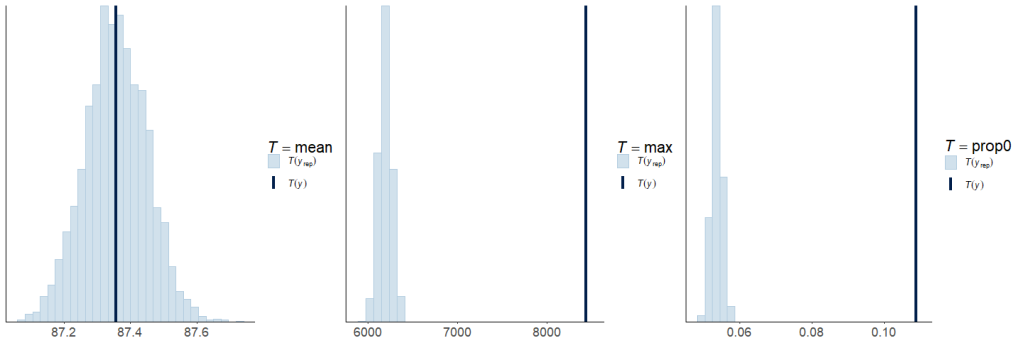
50	InstMon12	0.43	0.07	0.30	0.57	1.00
51	ProdMon2	−0.13	0.07	−0.26	0.01	1.00
52	ProdMon3	−0.19	0.08	−0.35	−0.03	1.01
53	ProdMon4	−0.15	0.06	−0.26	−0.03	1.00
54	ProdMon5	−0.10	0.08	−0.26	0.07	1.01
55	ProdMon6	−0.08	0.07	−0.22	0.05	1.00
56	ProdMon7	−0.14	0.08	−0.30	0.01	1.00
57	ProdMon8	−0.14	0.09	−0.31	0.04	1.00
58	ProdMon9	0.01	0.09	−0.18	0.19	1.01
59	ProdMon10	0.07	0.08	−0.09	0.23	1.00
60	ProdMon11	−0.11	0.09	−0.29	0.07	1.00
61	ProdMon12	0.08	0.09	−0.10	0.27	1.00

As in the chain diagnostic plots, all posterior predictive plots demonstrate the similarity between the Poisson and binomial models. This might be expected since as trial numbers become larger binomial distribution converges to Poisson distribution. For most of the observations in the data, n is quite large; thus, binomial distribution executes its limiting behavior.



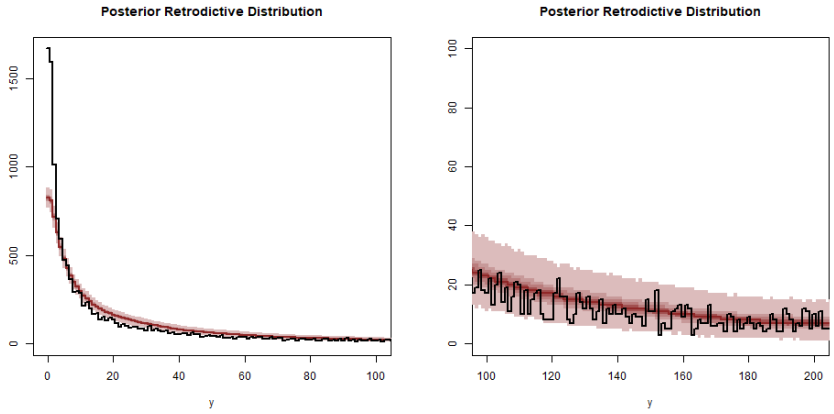
(a) $\widehat{R}$ values (b) Ratio of effective sample size

Figure 6.12: Chain diagnostics

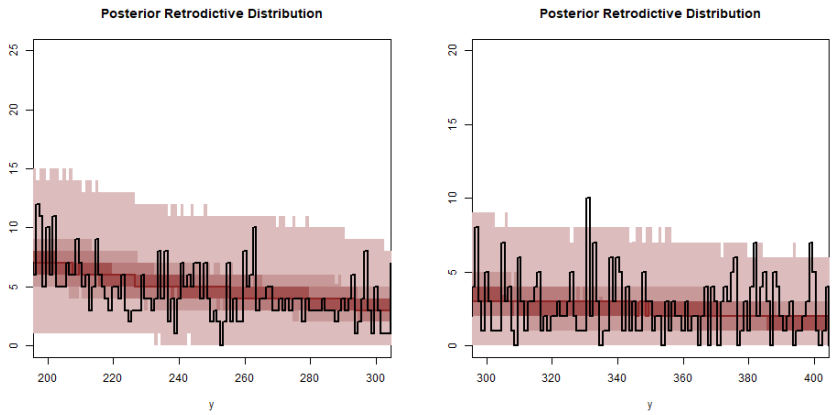


(a) Mean (b) Max (c) Zero portion

Figure 6.13: Posterior predictive check on mean, max, and zero Portion



(a) Interval [0,100] (b) Interval [100,200]



(c) Interval [200,300] (d) Interval [300,400]

Figure 6.14: Posterior retrodictive check

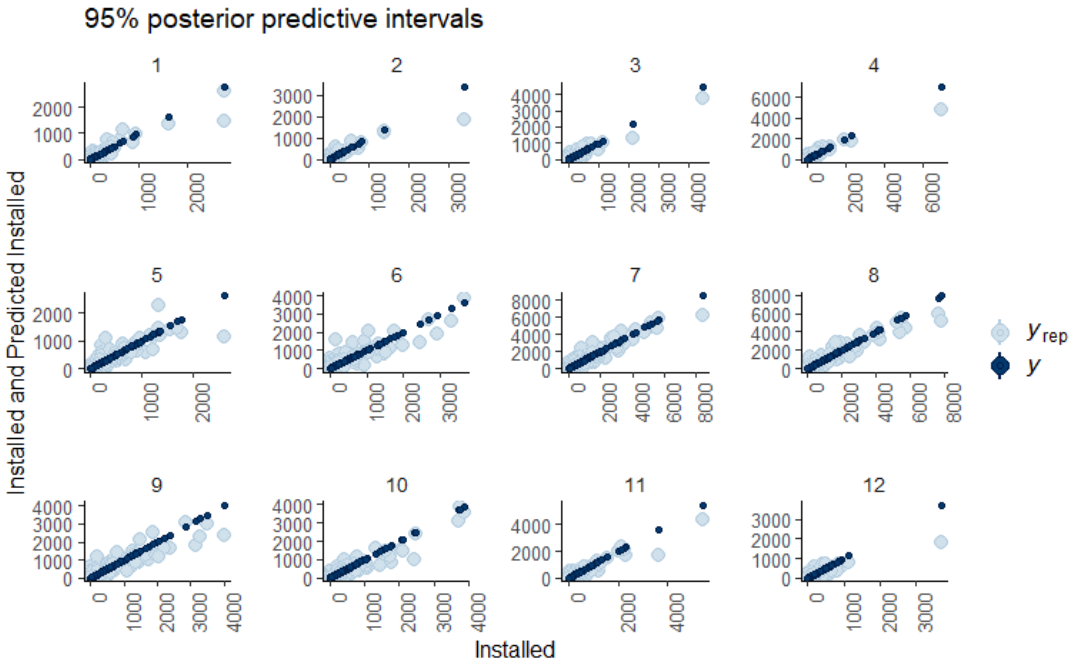


Figure 6.15: Posterior predictive distribution for each installation month

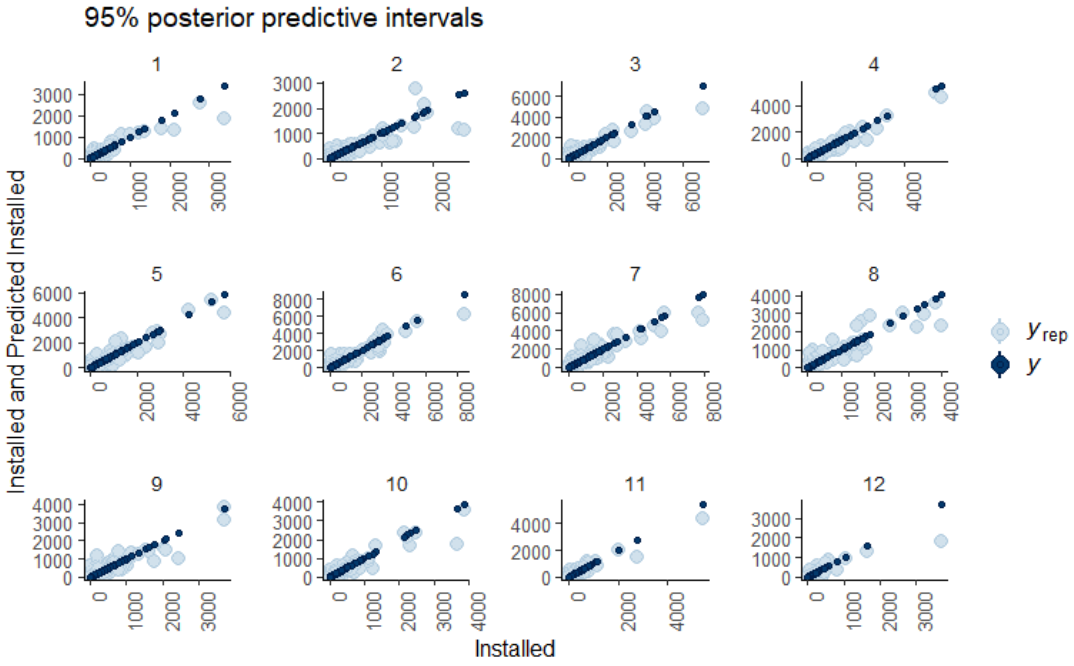


Figure 6.16: Posterior predictive distribution for each production month

6.2.4 Beta-Binomial Model Diagnostics

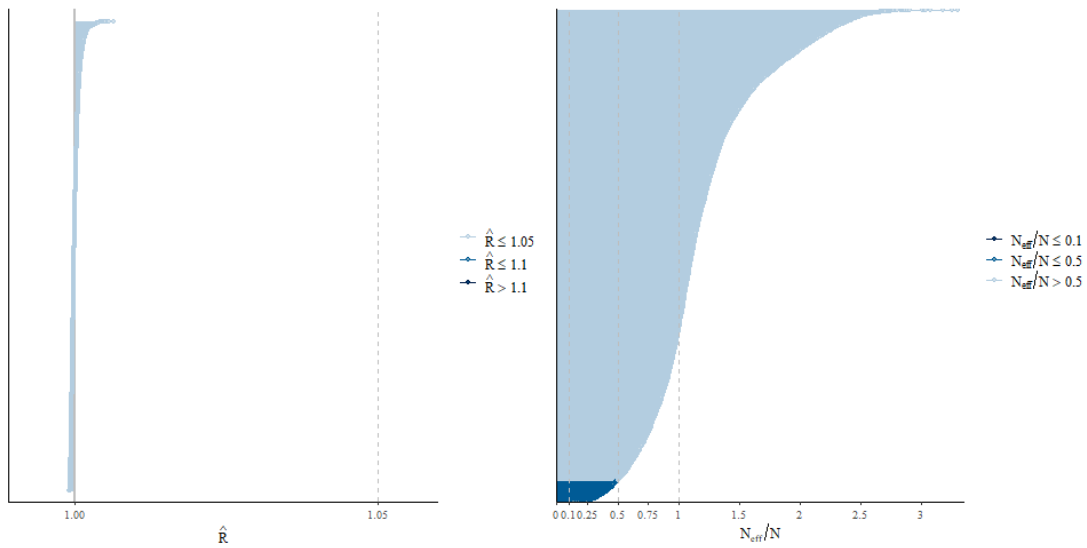
The model summary of the beta-binomial model is presented in the Listing 6.4. Similar to the other models, population-level installation month effects are more significant. Most of the population-level production month effects are insignificant. Similar to the negative binomial model, group-level effects are weak.

Listing 6.4: Summary of beta-binomial model

```
 Family: beta_binomial2
 Links: mu = logit; phi = identity
 Formula: Installed | trials(Produced) ~ 1 + InstMon + ProdMon
                    + (1 + InstMon + ProdMon | Model)
 Data: InstallationTrain (Number of observations: 15321)
 Draws: 4 chains, each with iter = 1000; warmup = 0; thin = 1;
          total post-warmup draws = 4000

 Group-Level Effects:
 ~Model (Number of levels: 141)
                 Estimate Est.Error l-95% CI u-95% CI Rhat
 sd(Intercept)       0.39      0.03      0.33      0.46 1.00
 sd(InstMon2)        0.06      0.04      0.00      0.17 1.00
 sd(InstMon3)        0.10      0.05      0.01      0.21 1.00
 sd(InstMon4)        0.20      0.05      0.10      0.30 1.00
 sd(InstMon5)        0.09      0.05      0.00      0.20 1.00
 sd(InstMon6)        0.16      0.05      0.05      0.25 1.00
 sd(InstMon7)        0.26      0.03      0.20      0.33 1.00
 sd(InstMon8)        0.42      0.04      0.35      0.50 1.00
 sd(InstMon9)        0.22      0.04      0.15      0.29 1.00
 sd(InstMon10)       0.15      0.05      0.05      0.24 1.00
 sd(InstMon11)       0.10      0.05      0.01      0.20 1.00
 sd(InstMon12)       0.24      0.04      0.16      0.33 1.00
 sd(ProdMon2)        0.33      0.05      0.23      0.43 1.00
 sd(ProdMon3)        0.29      0.05      0.20      0.38 1.00
 sd(ProdMon4)        0.22      0.05      0.13      0.31 1.00
```

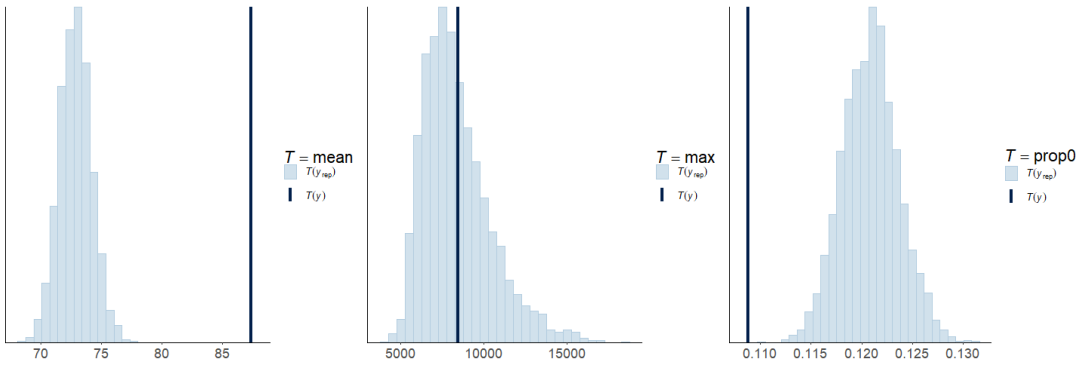
```
sd(ProdMon5)          0.26      0.05      0.18      0.36 1.00
sd(ProdMon6)          0.19      0.06      0.06      0.30 1.00
sd(ProdMon7)          0.28      0.05      0.19      0.38 1.00
sd(ProdMon8)          0.18      0.07      0.03      0.32 1.00
sd(ProdMon9)          0.29      0.05      0.19      0.40 1.00
sd(ProdMon10)         0.36      0.05      0.27      0.47 1.00
sd(ProdMon11)         0.21      0.06      0.09      0.32 1.00
sd(ProdMon12)         0.28      0.06      0.17      0.40 1.00

Population-Level Effects:
             Estimate Est.Error l-95% CI u-95% CI Rhat
Intercept       -2.99      0.05     -3.09     -2.89 1.00
InstMon2         0.15      0.05      0.06      0.24 1.00
InstMon3         0.39      0.05      0.30      0.48 1.00
InstMon4         0.32      0.05      0.22      0.41 1.00
InstMon5         0.64      0.04      0.56      0.73 1.00
InstMon6         0.84      0.05      0.76      0.93 1.00
InstMon7         1.21      0.05      1.12      1.30 1.00
InstMon8         1.14      0.06      1.03      1.25 1.00
InstMon9         0.80      0.05      0.71      0.89 1.00
InstMon10        0.52      0.04      0.43      0.61 1.00
InstMon11        0.30      0.04      0.21      0.38 1.00
InstMon12        0.25      0.05      0.15      0.34 1.00
ProdMon2        -0.08      0.05     -0.18      0.02 1.00
ProdMon3        -0.13      0.05     -0.22     -0.04 1.00
ProdMon4        -0.04      0.04     -0.12      0.05 1.00
ProdMon5        -0.06      0.04     -0.15      0.03 1.00
ProdMon6        -0.02      0.04     -0.10      0.06 1.00
ProdMon7        -0.07      0.04     -0.15      0.02 1.00
ProdMon8         0.01      0.05     -0.09      0.09 1.00
ProdMon9         0.09      0.05     -0.02      0.19 1.00
ProdMon10        0.08      0.06     -0.03      0.19 1.00
ProdMon11        0.03      0.05     -0.07      0.12 1.00
ProdMon12        0.09      0.05     -0.02      0.20 1.00
```

```
Family Specific Parameters:
     Estimate Est.Error l-95% CI u-95% CI Rhat
phi     15.33      0.23     14.88     15.79 1.00
```



(a) $\widehat{R}$ values (b) Ratio of effective sample size

Figure 6.17: Chain diagnostics

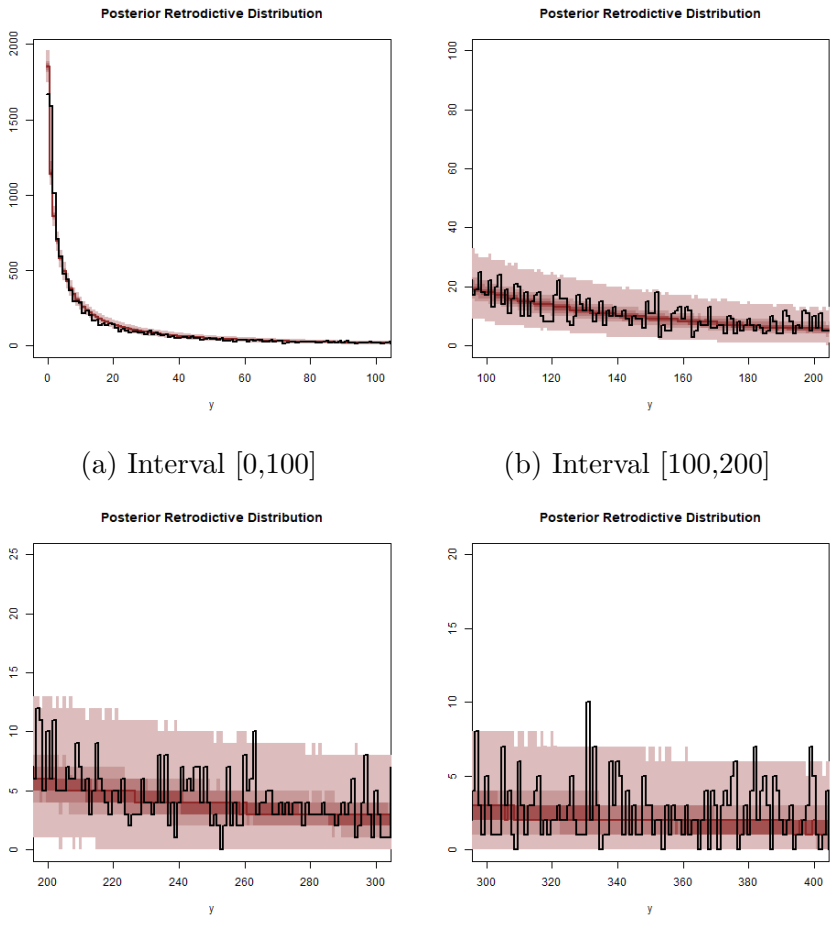


(a) Mean (b) Max (c) Zero portion

Figure 6.18: Posterior predictive check on mean, max, and zero portion

Figure 6.17 shows that chain diagnostics of the beta-binomial model are decent. Considering the previous plots, one can conclude that overdispersed models provide better sampling in this structure.

Although one may expect similarities between the negative binomial and the beta-binomial model, Figure 6.18 shows that posterior predictive distributions are slightly different. Unlike the negative binomial model, for this model the zero portion is over-estimated rather than under-estimated.



(a) Interval [0,100]

(b) Interval [100,200]

(c) Interval [200,300]

(d) Interval [300,400]

Figure 6.19: Posterior retrodictive check

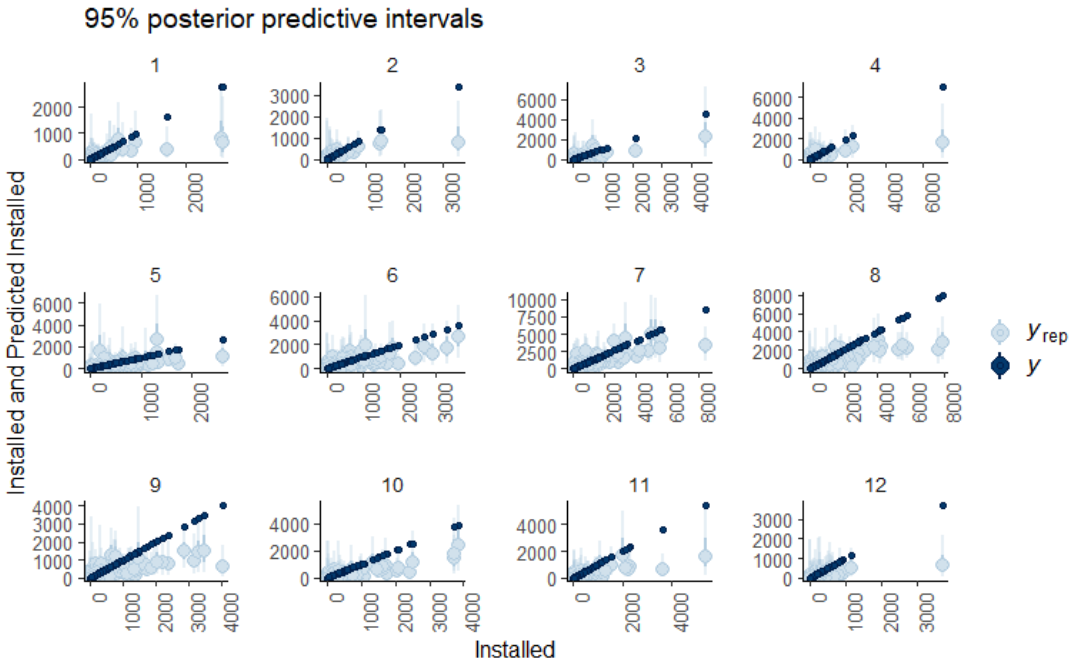


Figure 6.20: Posterior predictive distribution for each installation month

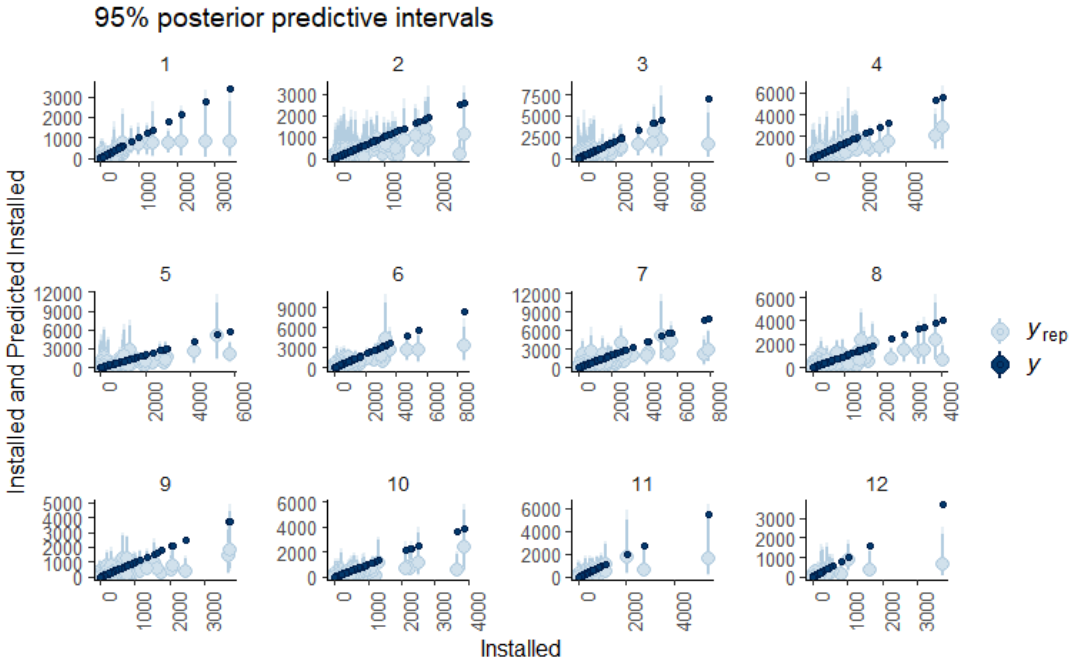


Figure 6.21: Posterior predictive distribution for each production month

Figure 6.19 indicates that the beta-binomial model has similar behavior to the negative binomial in terms of counts of y. Figures 6.20 and Figures 6.21 shows that beta-binomial under-estimates the larger installation numbers similar to the negative binomial model.

6.2.5 Negative Binomial Hurdle Model Diagnostics

Listing 6.5 shows the summary of the negative binomial hurdle model. Population-level installation month effects of July (InstMon7) and August (InstMon8) on positive counts are more sinificant than the effects of other months. On the other hand, population-level production month effects on positive counts are insignificant; most of the estimated parameters are very close to zero. Group-level effects of installation month and production month on positive counts are not very strong. This indicates that the refrigerator model doesn't influence population-level effects on positive counts.

Listing 6.5 shows that the population-level effects on zero counts are negative, meaning that they decrease the probability of zero installation. One can observe that the group-level effects of production month are strong. This indicates that the refrigerator model influences the effect of production month on zero counts. The estimated effect of refrigerator type on inverse-overdispersion parameter is also presented in Listing 6.5. One can notice that the effects of refrigerator types are quite different.

Listing 6.5: Summary of negative binomial hurdle model

```
1  Family: hurdle_negbinomial
2   Links: mu = log; shape = log; hu = logit
3 Formula: Installed ~ 1 + InstMon + ProdMon
4                        + offset(log(Produced)) + (1 + InstMon
5                        + ProdMon | Model)
6                  ZeroInstalled ~ 1 + InstMon + ProdMon
7                        + (1 + InstMon + ProdMon | Model)
8                  shape ~ Model
```

```
 Data: InstallationTrain (Number of observations: 15321)
Draws: 4 chains, each with iter = 1000; warmup = 0; thin = 1;
       total post-warmup draws = 4000

Group-Level Effects:
~Model (Number of levels: 141)
                     Estimate Est.Error l-95% CI u-95% CI Rhat
sd(Intercept)            0.44      0.04     0.37     0.51 1.00
sd(InstMon2)             0.21      0.05     0.11     0.31 1.00
sd(InstMon3)             0.13      0.06     0.02     0.23 1.01
sd(InstMon4)             0.18      0.05     0.07     0.28 1.00
sd(InstMon5)             0.12      0.05     0.02     0.21 1.01
sd(InstMon6)             0.14      0.05     0.03     0.23 1.01
sd(InstMon7)             0.23      0.04     0.16     0.30 1.01
sd(InstMon8)             0.35      0.04     0.29     0.42 1.00
sd(InstMon9)             0.22      0.04     0.16     0.30 1.00
sd(InstMon10)            0.18      0.04     0.08     0.26 1.00
sd(InstMon11)            0.16      0.05     0.05     0.25 1.00
sd(InstMon12)            0.26      0.04     0.17     0.34 1.00
sd(ProdMon2)             0.05      0.03     0.00     0.13 1.00
sd(ProdMon3)             0.06      0.04     0.00     0.15 1.00
sd(ProdMon4)             0.10      0.05     0.01     0.20 1.00
sd(ProdMon5)             0.20      0.05     0.10     0.31 1.01
sd(ProdMon6)             0.08      0.05     0.00     0.19 1.00
sd(ProdMon7)             0.15      0.04     0.06     0.23 1.00
sd(ProdMon8)             0.13      0.05     0.02     0.23 1.01
sd(ProdMon9)             0.22      0.04     0.14     0.31 1.00
sd(ProdMon10)            0.31      0.05     0.22     0.41 1.00
sd(ProdMon11)            0.14      0.06     0.02     0.25 1.00
sd(ProdMon12)            0.30      0.06     0.18     0.43 1.00
sd(zero_Intercept)       1.44      0.14     1.19     1.73 1.00
sd(zero_InstMon2)        0.31      0.18     0.01     0.68 1.01
sd(zero_InstMon3)        0.15      0.11     0.01     0.42 1.00
sd(zero_InstMon4)        0.18      0.13     0.01     0.49 1.00
sd(zero_InstMon5)        0.25      0.18     0.01     0.65 1.01
```

```
sd(zero_InstMon6)      0.15      0.12      0.01      0.44 1.00
sd(zero_InstMon7)      0.26      0.19      0.01      0.70 1.00
sd(zero_InstMon8)      0.56      0.24      0.08      1.02 1.00
sd(zero_InstMon9)      0.33      0.21      0.02      0.79 1.00
sd(zero_InstMon10)     0.21      0.15      0.01      0.55 1.00
sd(zero_InstMon11)     0.19      0.14      0.01      0.50 1.00
sd(zero_InstMon12)     1.11      0.85      0.04      3.17 1.00
sd(zero_ProdMon2)      1.31      0.18      0.97      1.69 1.00
sd(zero_ProdMon3)      1.23      0.22      0.85      1.69 1.00
sd(zero_ProdMon4)      1.12      0.21      0.74      1.59 1.00
sd(zero_ProdMon5)      1.37      0.24      0.94      1.85 1.00
sd(zero_ProdMon6)      1.04      0.26      0.56      1.60 1.00
sd(zero_ProdMon7)      1.15      0.29      0.62      1.76 1.00
sd(zero_ProdMon8)      2.07      0.37      1.45      2.87 1.00
sd(zero_ProdMon9)      2.07      0.43      1.36      2.98 1.00
sd(zero_ProdMon10)     1.38      0.31      0.83      2.05 1.00
sd(zero_ProdMon11)     1.44      0.37      0.79      2.25 1.00
sd(zero_ProdMon12)     1.28      0.31      0.72      1.99 1.00

Population-Level Effects:
                   Estimate Est.Error l-95% CI u-95% CI Rhat
Intercept            -3.07      0.06    -3.18    -2.96 1.01
shape_Intercept       0.37      0.21    -0.05     0.76 1.03
zero_Intercept       -0.87      0.18    -1.24    -0.51 1.00
InstMon2              0.06      0.05    -0.04     0.16 1.00
InstMon3              0.36      0.05     0.27     0.45 1.00
InstMon4              0.35      0.05     0.25     0.44 1.00
InstMon5              0.64      0.04     0.55     0.73 1.00
InstMon6              0.92      0.04     0.83     1.00 1.00
InstMon7              1.20      0.05     1.10     1.29 1.00
InstMon8              1.16      0.06     1.05     1.27 1.00
InstMon9              0.80      0.05     0.70     0.89 1.00
InstMon10             0.49      0.05     0.40     0.58 1.01
InstMon11             0.19      0.05     0.10     0.28 1.00
InstMon12             0.00      0.05    -0.10     0.10 1.00
```

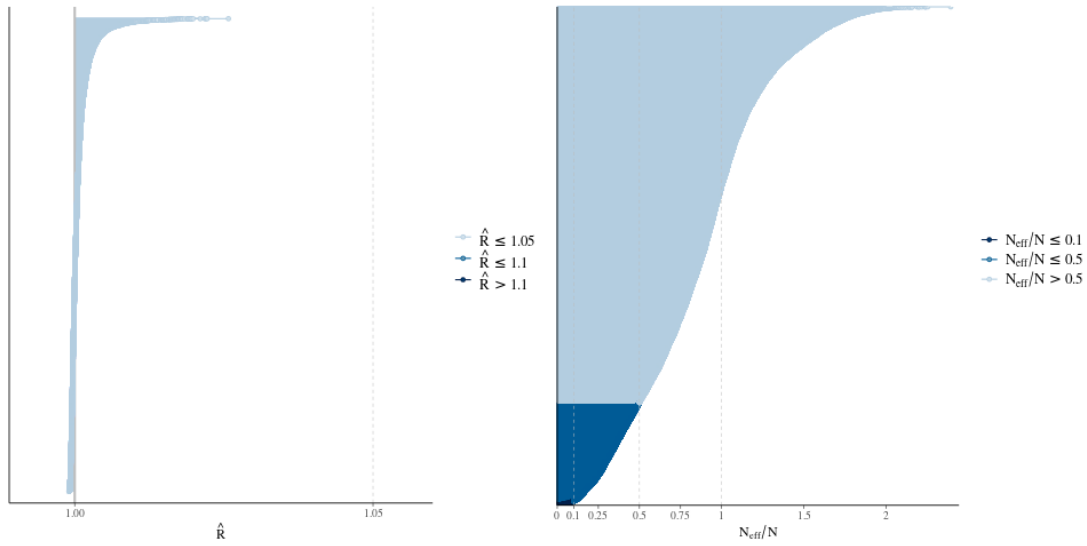
79	ProdMon2	−0.09	0.03	−0.16	−0.02	1.00
80	ProdMon3	−0.12	0.03	−0.19	−0.05	1.00
81	ProdMon4	−0.12	0.04	−0.19	−0.05	1.00
82	ProdMon5	−0.14	0.04	−0.23	−0.06	1.00
83	ProdMon6	−0.06	0.04	−0.13	0.02	1.00
84	ProdMon7	−0.07	0.04	−0.15	0.01	1.00
85	ProdMon8	0.03	0.04	−0.06	0.11	1.00
86	ProdMon9	0.09	0.05	0.01	0.19	1.00
87	ProdMon10	0.16	0.05	0.06	0.28	1.00
88	ProdMon11	0.02	0.04	−0.06	0.11	1.00
89	ProdMon12	0.11	0.06	−0.01	0.22	1.00
90	shape_Modela1M8f	3.49	4.09	−2.45	13.56	1.00
91	shape_Modela10M8e	4.83	3.93	0.31	14.62	1.00
92	shape_Modela11M8b	4.22	3.84	−0.90	13.85	1.00
93	shape_Modela2M8b	0.21	0.30	−0.37	0.78	1.01
94	shape_Modela3M8b	0.25	0.29	−0.33	0.82	1.01
95	shape_Modela6M3b	0.58	1.04	−1.09	2.85	1.00
96	shape_Modela7M8b	0.42	0.54	−0.66	1.45	1.00
97	shape_Modela8M3b	1.07	0.62	−0.09	2.35	1.00
98	shape_Modela9M8b	3.51	3.90	−1.32	13.33	1.00
99	shape_Modelaa1M1a	−1.65	0.29	−2.23	−1.09	1.01
100	shape_Modelaa1M1d	−0.21	0.26	−0.70	0.32	1.02
101	shape_Modelab1M1c	−0.01	0.24	−0.48	0.46	1.02
102	shape_Modelab1M4c	−0.28	0.99	−2.71	1.15	1.01
103	shape_Modelab1M6d	3.26	4.26	−3.83	13.75	1.00
104	shape_Modelab2M1a	−0.98	0.29	−1.54	−0.41	1.02
105	shape_Modelab2M1d	−1.88	0.42	−2.80	−1.13	1.01
106	shape_Modelab2M2f	−2.42	1.12	−5.09	−0.85	1.00
107	shape_Modelab3M4c	−0.22	0.27	−0.75	0.31	1.02
108	shape_Modelac2M4c	0.29	0.26	−0.22	0.81	1.02
109	shape_Modelad1M1c	−0.46	0.55	−1.64	0.51	1.00
110	shape_Modelba1M1b	−0.12	0.24	−0.57	0.36	1.02
111	shape_Modelba1M3a	0.78	0.52	−0.26	1.78	1.00
112	shape_Modelba1M3c	−0.02	0.26	−0.52	0.49	1.02
113	shape_Modelba10M5a	−0.62	0.26	−1.15	−0.11	1.02

114	shape_Modelba10M6c	0.48	0.32	−0.14	1.11	1.01
115	shape_Modelba11M4a	−0.26	0.26	−0.76	0.24	1.02
116	shape_Modelba12M1b	0.96	0.88	−0.19	2.27	1.00
117	shape_Modelba12M3a	0.07	0.60	−1.16	1.20	1.00
118	shape_Modelba13M1b	5.09	3.89	0.47	15.14	1.00
119	shape_Modelba14M4b	0.89	0.28	0.36	1.43	1.01
120	shape_Modelba15M3a	2.73	4.48	−4.44	13.78	1.00
121	shape_Modelba15M4b	1.25	0.32	0.64	1.90	1.01
122	shape_Modelba2M1b	−0.36	0.24	−0.82	0.14	1.02
123	shape_Modelba2M3a	0.18	0.41	−0.65	0.95	1.01
124	shape_Modelba2M3c	0.07	0.27	−0.45	0.60	1.02
125	shape_Modelba3M1b	0.34	0.24	−0.12	0.84	1.01
126	shape_Modelba3M3a	0.09	0.26	−0.39	0.61	1.02
127	shape_Modelba3M3c	0.01	0.36	−0.73	0.72	1.01
128	shape_Modelba3M4b	−0.03	0.25	−0.51	0.49	1.02
129	shape_Modelba4M3c	−0.33	0.26	−0.83	0.18	1.02
130	shape_Modelba4M6e	0.04	0.23	−0.41	0.52	1.02
131	shape_Modelba5M2d	0.06	0.43	−0.80	0.87	1.01
132	shape_Modelba5M4a	0.29	0.22	−0.14	0.75	1.02
133	shape_Modelba6M1b	−1.56	0.70	−3.17	−0.41	1.00
134	shape_Modelba6M3a	0.01	0.23	−0.42	0.47	1.02
135	shape_Modelba6M3c	0.22	0.27	−0.29	0.76	1.01
136	shape_Modelba7M3c	−0.76	1.89	−5.37	1.98	1.01
137	shape_Modelba7M4b	−0.68	0.24	−1.15	−0.17	1.02
138	shape_Modelba7M6c	0.69	0.29	0.13	1.26	1.01
139	shape_Modelba8M3c	0.16	0.49	−0.84	1.07	1.01
140	shape_Modelba8M4b	−0.19	0.30	−0.77	0.40	1.01
141	shape_Modelba9M2f	0.77	0.30	0.19	1.35	1.01
142	shape_Modelba9M5a	−0.10	0.23	−0.54	0.36	1.02
143	shape_Modelbb1M1b	−0.10	0.26	−0.63	0.42	1.02
144	shape_Modelbb1M3a	−0.88	0.38	−1.69	−0.19	1.01
145	shape_Modelbb1M3c	−1.18	0.47	−2.21	−0.34	1.01
146	shape_Modelbb10M4b	−0.45	0.30	−1.04	0.13	1.01
147	shape_Modelbb11M5a	−0.48	0.27	−1.01	0.06	1.02
148	shape_Modelbb13M3a	−0.78	0.79	−2.45	0.61	1.00

149	shape_Modelbb13M3c	−2.14	0.97	−4.25	−0.43	1.00
150	shape_Modelbb13M4b	−0.47	0.63	−1.75	0.68	1.00
151	shape_Modelbb14M3d	0.49	0.23	0.04	0.96	1.02
152	shape_Modelbb14M6c	0.63	0.24	0.17	1.11	1.02
153	shape_Modelbb15M3d	1.05	0.26	0.57	1.58	1.02
154	shape_Modelbb16M3d	1.11	0.33	0.48	1.78	1.01
155	shape_Modelbb17M3a	0.91	0.30	0.34	1.49	1.02
156	shape_Modelbb17M3c	0.58	0.28	0.03	1.13	1.01
157	shape_Modelbb18M3d	−0.81	2.34	−4.70	4.73	1.00
158	shape_Modelbb18M4a	−0.03	0.28	−0.58	0.53	1.01
159	shape_Modelbb19M1b	4.28	3.42	0.03	12.91	1.00
160	shape_Modelbb19M3a	−0.99	0.44	−1.89	−0.17	1.01
161	shape_Modelbb19M4a	1.23	0.61	0.21	2.51	1.00
162	shape_Modelbb2M3a	0.53	0.25	0.05	1.02	1.02
163	shape_Modelbb2M3c	0.11	0.43	−0.75	0.94	1.01
164	shape_Modelbb20M4a	0.91	0.33	0.28	1.57	1.01
165	shape_Modelbb21M4a	0.48	0.36	−0.25	1.17	1.01
166	shape_Modelbb22M4a	0.47	0.35	−0.21	1.14	1.01
167	shape_Modelbb3M3c	−1.47	1.23	−4.69	0.23	1.00
168	shape_Modelbb3M3d	1.11	0.31	0.48	1.73	1.01
169	shape_Modelbb3M6c	0.58	0.25	0.10	1.07	1.02
170	shape_Modelbb8M4a	−0.05	0.24	−0.50	0.43	1.02
171	shape_Modelbb9M3d	0.92	0.26	0.43	1.43	1.02
172	shape_Modelbb9M3e	0.13	0.27	−0.40	0.66	1.01
173	shape_ModelbcM3c	1.63	0.49	0.67	2.61	1.00
174	shape_Modelca2M2a	1.90	3.69	−2.33	11.20	1.00
175	shape_Modelcb1M3b	0.72	0.23	0.29	1.17	1.02
176	shape_Modelcb2M3b	0.62	0.23	0.17	1.09	1.02
177	shape_Modelcb3M3b	0.20	0.85	−1.68	1.73	1.00
178	shape_Modelcb4M3b	0.94	1.80	−0.99	6.12	1.00
179	shape_Modelcc1M3b	0.04	0.23	−0.40	0.52	1.02
180	shape_Modelcc2M3b	−0.35	0.27	−0.90	0.19	1.02
181	shape_Modelcc3M3b	0.90	0.27	0.39	1.43	1.02
182	shape_Modelda1M4a	0.45	0.23	0.01	0.93	1.02
183	shape_Modelda2M4a	0.90	0.24	0.44	1.39	1.02

184	shape_Modelda4M3a	0.83	0.28	0.28	1.38	1.02
185	shape_Modelda4M4a	1.05	0.27	0.54	1.58	1.02
186	shape_Modeldb1M3a	0.07	0.24	−0.39	0.56	1.02
187	shape_Modeldb1M4a	0.80	0.30	0.21	1.35	1.01
188	shape_Modeldb2M3a	0.49	0.24	0.02	0.98	1.02
189	shape_Modeldb2M4a	0.70	0.42	−0.11	1.49	1.01
190	shape_Modeldb2M8c	0.63	0.39	−0.14	1.40	1.01
191	shape_Modeldb3M4a	1.04	0.47	0.07	1.95	1.00
192	shape_Modelea1M5a	−0.80	0.26	−1.30	−0.30	1.02
193	shape_Modelea2M5a	0.03	0.26	−0.47	0.53	1.02
194	shape_Modeleb1M6a	0.09	0.24	−0.35	0.56	1.03
195	shape_Modeleb2M6a	0.01	0.29	−0.54	0.57	1.02
196	shape_Modeleb3M2f	−0.80	0.43	−1.72	−0.06	1.01
197	shape_Modeleb3M5a	0.29	0.25	−0.20	0.80	1.02
198	shape_Modeleb4M3e	−0.66	0.56	−1.97	0.24	1.01
199	shape_Modeleb4M6a	0.01	0.24	−0.45	0.50	1.02
200	shape_Modeleb5M5a	0.42	0.30	−0.17	1.01	1.01
201	shape_Modeleb6M2f	1.36	1.75	−1.01	6.35	1.00
202	shape_Modeleb6M5a	0.98	0.27	0.45	1.52	1.01
203	shape_Modelec1M3d	−0.57	0.72	−2.33	0.53	1.00
204	shape_Modelec1M3e	−0.29	3.80	−7.57	8.33	1.00
205	shape_Modelec1M5a	−0.20	0.25	−0.69	0.31	1.02
206	shape_Modelec3M3d	0.34	5.04	−8.72	11.04	1.00
207	shape_Modelec3M8c	0.76	0.29	0.19	1.34	1.02
208	shape_Modeled2M1a	−1.26	0.61	−2.48	−0.13	1.00
209	shape_Modeled2M1d	−1.13	0.62	−2.41	0.02	1.00
210	shape_Modeled3M1a	3.91	3.61	−0.38	13.52	1.00
211	shape_Modelee1M5a	−1.45	0.68	−2.86	−0.23	1.01
212	shape_Modelee1M6b	−0.92	1.07	−3.16	0.95	1.00
213	shape_Modelee1M6e	0.30	0.24	−0.17	0.80	1.02
214	shape_Modelee2M2g	0.57	0.52	−0.46	1.49	1.00
215	shape_Modelfa1M3b	0.29	0.25	−0.19	0.79	1.02
216	shape_Modelfa2M3b	−0.16	0.23	−0.60	0.29	1.02
217	shape_Modelfa3M3b	0.35	0.31	−0.28	0.96	1.01
218	shape_Modelfa4M3b	0.29	0.39	−0.50	1.03	1.01

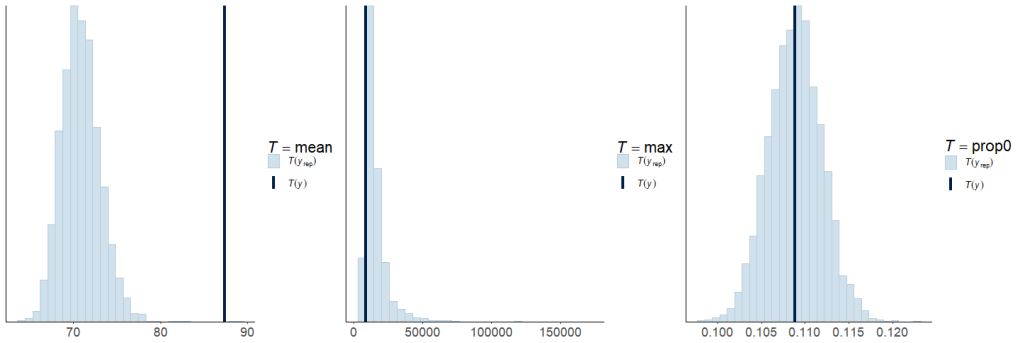
219	shape_ModelfbM8c	−1.68	1.22	−4.36	0.42	1.00
220	shape_Modelga1M2b	−1.58	0.30	−2.20	−1.00	1.01
221	shape_Modelga1M7a	−0.78	0.25	−1.28	−0.29	1.02
222	shape_Modelgb1M1a	−0.27	0.44	−1.14	0.56	1.00
223	shape_Modelgb1M1d	0.26	0.53	−0.85	1.33	1.00
224	shape_Modelha1M3d	−2.85	0.68	−4.33	−1.66	1.00
225	shape_Modelha1M3e	3.91	4.30	−2.45	14.30	1.00
226	shape_Modelha1M8a	0.29	0.69	−1.02	1.76	1.00
227	shape_Modelha3M4c	−0.49	0.61	−1.85	0.55	1.00
228	shape_Modelja1M3b	−0.28	0.78	−1.86	1.26	1.00
229	shape_Modelja2M3b	−1.06	0.65	−2.45	0.08	1.00
230	zero_InstMon2	−0.58	0.14	−0.86	−0.32	1.00
231	zero_InstMon3	−0.93	0.14	−1.21	−0.66	1.00
232	zero_InstMon4	−0.90	0.13	−1.17	−0.64	1.00
233	zero_InstMon5	−1.25	0.14	−1.52	−0.98	1.00
234	zero_InstMon6	−1.43	0.14	−1.71	−1.17	1.00
235	zero_InstMon7	−2.14	0.16	−2.46	−1.84	1.00
236	zero_InstMon8	−2.07	0.19	−2.47	−1.73	1.00
237	zero_InstMon9	−1.68	0.16	−2.01	−1.37	1.00
238	zero_InstMon10	−1.48	0.14	−1.76	−1.20	1.00
239	zero_InstMon11	−1.26	0.13	−1.53	−1.00	1.00
240	zero_InstMon12	−5.21	0.99	−7.87	−3.98	1.00
241	zero_ProdMon2	−0.28	0.21	−0.71	0.12	1.00
242	zero_ProdMon3	−0.44	0.23	−0.90	−0.02	1.00
243	zero_ProdMon4	−0.42	0.21	−0.85	−0.02	1.00
244	zero_ProdMon5	−0.68	0.24	−1.18	−0.24	1.00
245	zero_ProdMon6	−0.80	0.25	−1.33	−0.34	1.00
246	zero_ProdMon7	−1.13	0.27	−1.68	−0.65	1.00
247	zero_ProdMon8	−1.26	0.41	−2.13	−0.55	1.00
248	zero_ProdMon9	−1.59	0.44	−2.54	−0.83	1.00
249	zero_ProdMon10	−0.77	0.29	−1.41	−0.23	1.00
250	zero_ProdMon11	−1.01	0.36	−1.79	−0.40	1.00
251	zero_ProdMon12	−0.32	0.27	−0.92	0.18	1.00



(a) $\widehat{R}$ values (b) Ratio of effective sample size

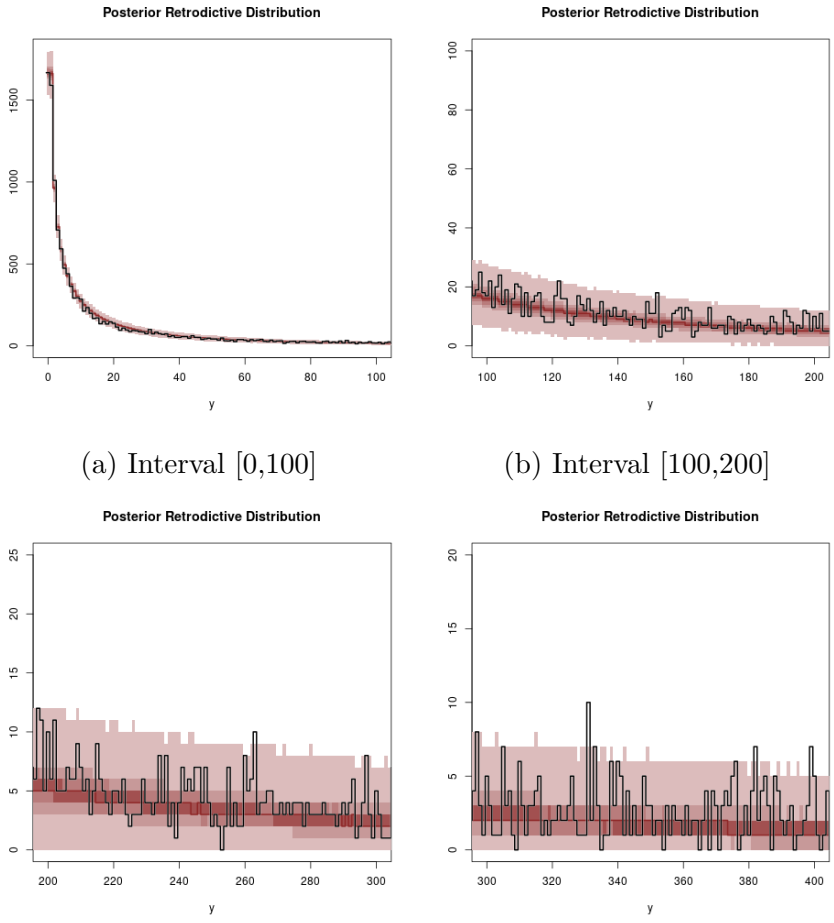
Figure 6.22: Chain diagnostics

The model's chain diagnostic plots are reasonable but Figure 6.22b shows that there are a few inefficient samples with n_eff ratio smaller than 0.1. Some of the posterior predictive plots are improved compared to other overdispersed models. Figure 6.23c shows that the zero portion in posterior predictive distribution is fairly centered around the zero portion in the data. This is achieved by modeling the zeroes with another distribution. The hurdle model also can capture maximum values in the data. This is also expected since the negative binomial distribution is used for modeling positive integers in the data. Unfortunately, Figure 6.23a shows that the mean distribution of samples is far from the true mean .



(a) Mean (b) Max (c) Zero portion

Figure 6.23: Posterior predictive check on mean, max, and zero portion



(a) Interval [0,100] (b) Interval [100,200]

(c) Interval [200,300] (d) Interval [300,400]

Figure 6.24: Posterior retrodictive check

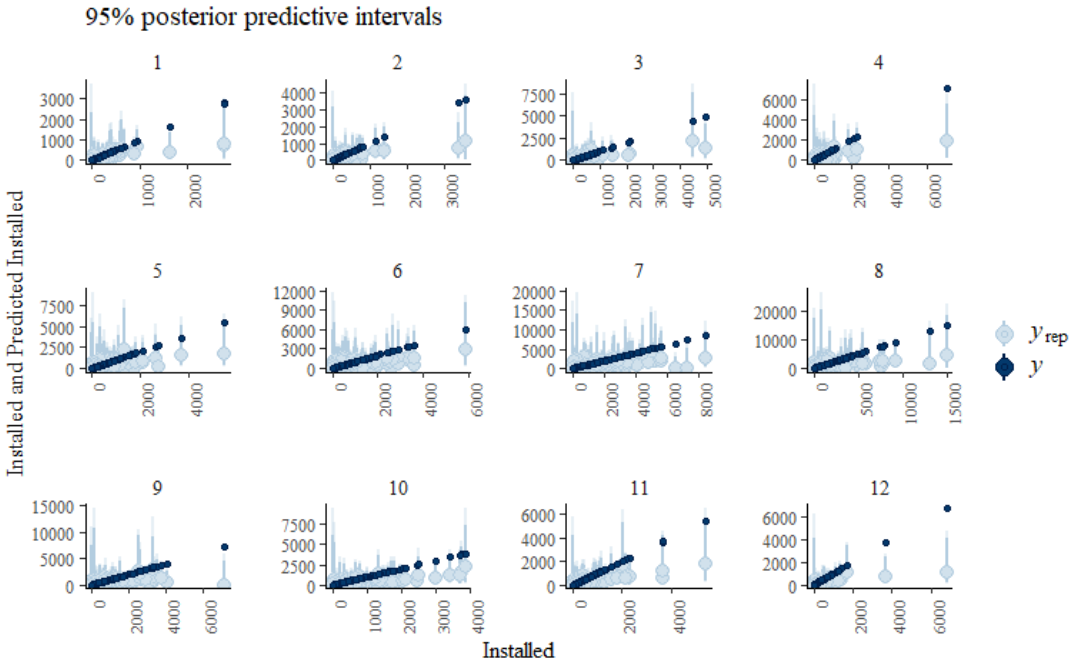


Figure 6.25: Posterior predictive distribution for each installation month

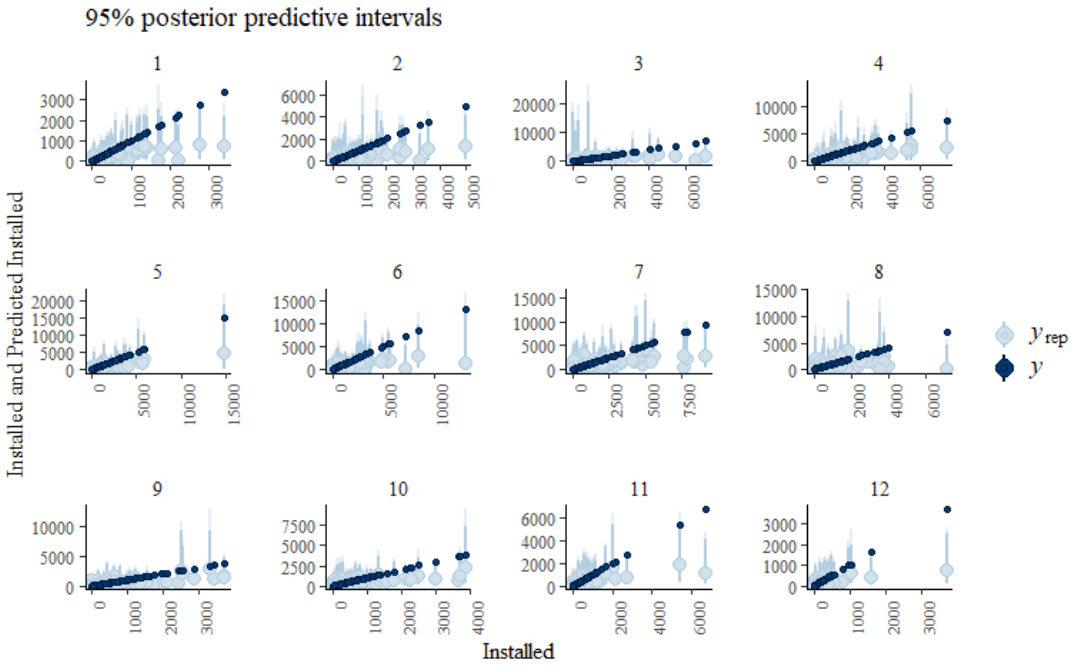


Figure 6.26: Posterior predictive distribution for each production month

Figure 6.24a shows that the negative binomial hurdle model is better than other models in terms of estimating the frequencies of small y values. But for larger values, its estimations are similar to the negative binomial model. Posterior predictive interval plots in Figures 6.25 and 6.26 shows that predictions are still not very accurate.

Here, we presented the explicit form and of the selected model for estimating the number of the installed refrigerator. We decided to use the negative binomial hurdle model for estimating installation numbers. A detailed discussion on selecting final models is made in Chapter 8. In the explicit form of the model, we denote the grouping factor, namely refrigerator and compressor model, with m.

$$
\begin{aligned}
y^m_{\text{installed}} \mid y^m_{\text{installed}} > 0 &\sim \text{NB}\left(\mu^m, \alpha\right), \\
\mathbb{P}\left(y^m_{\text{installed}} = 0\right) &= \pi^m, \\
\text{logit}\left(\pi^m\right) &= \gamma_0 + \gamma_1^m \text{InstMon2} + \gamma_2^m \text{InstMon3} + \cdots + \gamma_{11}^m \text{InstMon12} \\
&\quad + \gamma_{12}^m \text{ProdMon2} + \gamma_{13}^m \text{ProdMon3} + \cdots + \gamma_{22}^m \text{ProdMon12}, \\
\log\left(\mu^m\right) &= \beta_0 + \beta_1^m \text{InstMon2} + \beta_2^m \text{InstMon3} + \cdots + \beta_{11}^m \text{InstMon12} \\
&\quad + \beta_{12}^m \text{ProdMon2} + \beta_{13}^m \text{ProdMon3} + \cdots + \beta_{22}^m \text{ProdMon12} \\
&\quad + \log\left(n^m_{\text{produced}}\right), \\
\log\left(\alpha\right) &= \theta_0 + \theta_1 \text{Model2} + \ldots \theta_{140} \text{Model141} \\
\begin{bmatrix} \gamma_0^m \\ \gamma_1^m \\ \vdots \\ \gamma_{22}^m \end{bmatrix} &\sim \mathcal{N}\left(\begin{bmatrix} \bar{\gamma}_0 \\ \bar{\gamma}_1 \\ \vdots \\ \bar{\gamma}_{22} \end{bmatrix}, \Sigma^\pi \right), \\
\begin{bmatrix} \beta_0^m \\ \beta_1^m \\ \vdots \\ \beta_{22}^m \end{bmatrix} &\sim \mathcal{N}\left(\begin{bmatrix} \bar{\beta}_0 \\ \bar{\beta}_1 \\ \vdots \\ \bar{\beta}_{22} \end{bmatrix}, \Sigma^\mu \right),
\end{aligned}
$$

$$
\begin{aligned}
\Sigma^{\pi} &= \begin{bmatrix} \sigma_0^{\pi} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_{22}^{\pi} \end{bmatrix} R^{\pi} \begin{bmatrix} \sigma_0^{\pi} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_{22}^{\pi} \end{bmatrix}, \\
\Sigma^{\mu} &= \begin{bmatrix} \sigma_0^{\mu} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_{22}^{\mu} \end{bmatrix} R^{\mu} \begin{bmatrix} \sigma_0^{\mu} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_{22}^{\mu} \end{bmatrix}, \\
\theta_0, \theta_1, \ldots, \theta_{140} &\overset{\text{iid}}{\sim} t_7\left(0, 10\right), \\
\bar{\gamma}_0, \bar{\gamma}_1, \ldots, \bar{\gamma}_{22} &\overset{\text{iid}}{\sim} t_3\left(0, 10\right), \\
\bar{\beta}_0, \bar{\beta}_1, \ldots, \bar{\beta}_{22} &\overset{\text{iid}}{\sim} t_3\left(0, 10\right), \\
\sigma_0^{\pi}, \sigma_1^{\pi}, \ldots, \sigma_{22}^{\pi} &\overset{\text{iid}}{\sim} C^{+}\left(0, 3\right), \\
\sigma_0^{\mu}, \sigma_1^{\mu}, \ldots, \sigma_{22}^{\mu} &\overset{\text{iid}}{\sim} C^{+}\left(0, 3\right), \\
L^{\pi} &\sim \text{LKJCorr}\left(2\right), \\
L^{\mu} &\sim \text{LKJCorr}\left(2\right).
\end{aligned}
\tag{6.6}
$$

Chapter 7

Hierarchical Bayesian Model for Prediction of the Number of In-service Refrigerator Failures

In this chapter, structure of the models and their diagnostic plots are presented. The same models in Chapter 6 are used with little differences. The covariate age was transformed into a binary variable that indicates whether the refrigerator broke down in the first month after being sold. Then it was added to the models as another covariate. As another difference, Poisson distribution is used in the hurdle model.

7.1 Models

In this chapter, the structures of the models are presented. The general structure of the models is shown with a DAG in Figure 7.1.

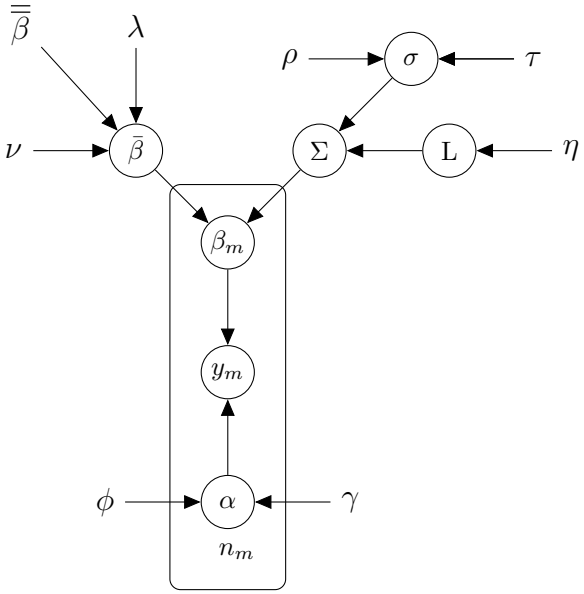


Figure 7.1: DAG for hierarchical models

7.1.1 Poisson Model

In this model, we used the Poisson Likelihood function to model the number of in-service refrigerator failures. The structure of the model can be defined as

$$
\begin{aligned}
y_m &\sim \text{Pois}\left(\mu_m\right), \\
\mu_m &= \exp\left(X_m\beta_m + \log\left(n_m\right)\right), \\
\beta_m &\sim \mathcal{N}(\bar{\beta}, \Sigma), \\
\bar{\beta} &\sim t_3\left(0, 10\right), \\
\Sigma &= \sigma L \sigma^T, \\
L &\sim \text{LKJCorr}\left(2\right), \\
\sigma &\sim C^+\left(0, 3\right).
\end{aligned}
\tag{7.1}
$$

7.1.2 Negative Binomial Model

In this model, we used the Negative Binomial Likelihood function to model the number of in-service refrigerator failures. The structure of the model can be defined as

$$\begin{aligned} y_m &\sim \mathrm{NB}\left(\mu_m, \alpha\right), \\ \mu_m &= \exp\left(X_m \beta_m + \log\left(n_m\right)\right), \\ \beta_m &\sim \mathcal{N}(\bar{\beta}, \Sigma), \\ \bar{\beta} &\sim t_3\left(0, 10\right), \\ \Sigma &= \sigma L \sigma^T, \\ L &\sim \mathrm{LKJCorr}\left(2\right), \\ \sigma &\sim C^+\left(0, 3\right), \\ \alpha &\sim \mathrm{Gamma}\left(0.01, 0.01\right). \end{aligned} \tag{7.2}$$

7.1.3 Binomial Model

In this model, we used the binomial likelihood function to model the number of in-service refrigerator failures. The structure of the model can be defined as

$$\begin{aligned} y_m &\sim B\left(n_m, \mu_m\right), \\ \mathrm{logit}\left(\mu_m\right) &= X_m \beta_m, \\ \beta_m &\sim \mathcal{N}(\bar{\beta}, \Sigma), \\ \bar{\beta} &\sim t_3\left(0, 10\right), \\ \Sigma &= \sigma L \sigma^T, \\ L &\sim \mathrm{LKJCorr}\left(2\right), \\ \sigma &\sim C^+\left(0, 3\right). \end{aligned} \tag{7.3}$$

7.1.4 Beta-Binomial Model

In this model, we used the beta-binomial likelihood function. The structure of the beta-binomial model can be defined as

$$\begin{aligned} y_m &\sim \text{BetaBin}\left(n_m, \mu_m, \alpha\right), \\ \text{logit}\left(\mu_m\right) &= X_m \beta_m, \\ \beta_m &\sim \mathcal{N}(\bar{\beta}, \Sigma), \\ \bar{\beta} &\sim t_3\left(0, 10\right), \\ \Sigma &= \sigma L \sigma^T, \\ L &\sim \text{LKJCorr}\left(2\right), \\ \sigma &\sim C^+\left(0, 3\right), \\ \alpha &\sim \text{Gamma}\left(0.01, 0.01\right). \end{aligned} \tag{7.4}$$

7.1.5 Poisson Hurdle Model

In this model, we use two different likelihood functions to express positive and zero counts separately. The first part of the model has a binomial likelihood function that decides whether the response will be equal to zero or not. The second part models the positive counts, and we used a truncated Poisson likelihood function for this purpose. We select Poisson distribution because we believe that data is not overdispersed. The structure of the model can be defined as

$$\begin{aligned} y_m \mid y_m > 0 &\sim \text{Pois}\left(\mu_m\right), \\ \mathbb{P}\left(y_m = 0\right) &= \pi_m, \\ \text{logit}\left(\pi_m\right) &= X_m \gamma_m, \\ \log\left(\mu_m\right) &= X_m \beta_m + \log\left(n_m\right), \\ \gamma_m &\sim \mathcal{N}(\bar{\gamma}, \Sigma_\pi), \\ \beta_m &\sim \mathcal{N}(\bar{\beta}, \Sigma_\mu), \\ \bar{\gamma} &\sim t_3\left(0, 10\right), \\ \bar{\beta} &\sim t_3\left(0, 10\right), \end{aligned} \tag{7.5}$$

$$
\begin{aligned}
\Sigma_{\pi} &= \sigma_{\pi} L_{\pi} \sigma_{\pi}^{T}, \\
L_{\pi} &\sim \mathrm{LKJCorr}\left(2\right), \\
\sigma_{\pi} &\sim C^{+}\left(0, 3\right), \\
\Sigma_{\mu} &= \sigma_{\mu} L_{\mu} \sigma_{\mu}^{T}, \\
L_{\mu} &\sim \mathrm{LKJCorr}\left(2\right), \\
\sigma_{\mu} &\sim C^{+}\left(0, 3\right).
\end{aligned}
$$

7.2 Diagnostics

In this section, the summary and the diagnostics of models are presented. Variables that are used in model summaries are described in Table 2.1.

7.2.1 Poisson Model Diagnostics

Listing 7.1 shows the summary of the Poisson model. Besides installation month and production month, the age turned into a binary covariate that describes wheater failure occurs in the first month after it is sold and added to the model.

All the population effects are negative, meaning that refrigerators manufactured and installed in January have the highest failure rate. On the other hand, the population-level effects of some production months are insignificant, meaning that they have the same failure rate as the reference level (January). Although group-level effects are not very strong, some allow positive production effects. So, some production months can increase the failure probability for some models. On the other hand, the effect of age is always negative, meaning that refrigerators with an age larger than one have a lower failure rate.

Listing 7.1: Summary of Poisson model

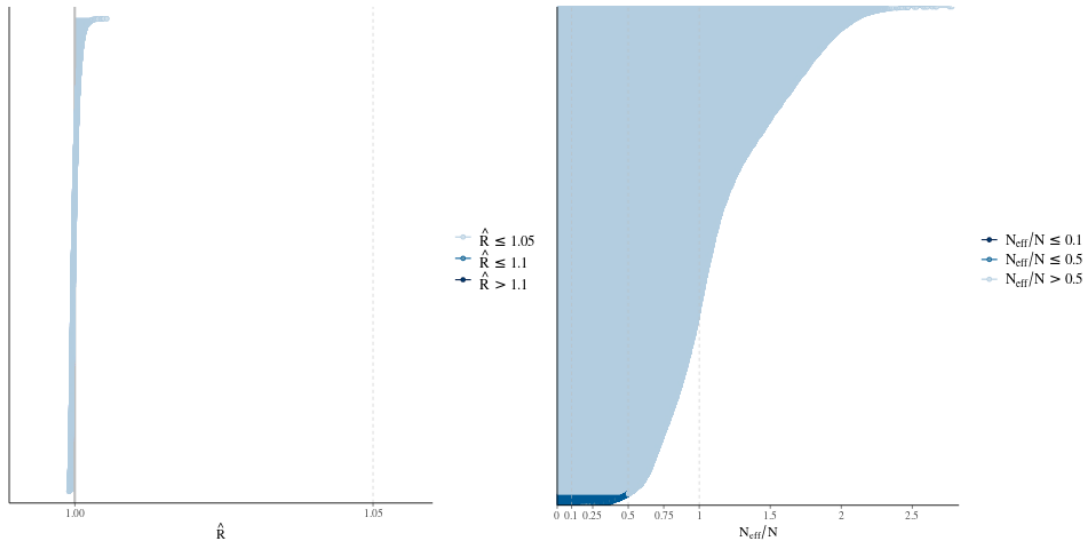
```
 Family: poisson
 Links: mu = log
 Formula: Failure ~ 1 + AgeFailBinary + InstMon + ProdMon
                  + offset(log(Installed))
                  + (1 + AgeFailBinary + InstMon
                  + ProdMon | Model)
 Data: FailureTrain (Number of observations: 16464)
 Draws: 4 chains, each with iter = 1000; warmup = 0; thin = 1;
          total post-warmup draws = 4000

 Group-Level Effects:
 ~Model (Number of levels: 141)
                  Estimate Est.Error l-95% CI u-95% CI Rhat
 sd(Intercept)        0.80      0.08      0.66      0.95 1.00
 sd(Age>1)            0.34      0.04      0.26      0.43 1.00
 sd(InstMon02)        0.29      0.11      0.08      0.51 1.00
 sd(InstMon03)        0.17      0.09      0.01      0.37 1.00
 sd(InstMon04)        0.08      0.07      0.00      0.24 1.00
 sd(InstMon05)        0.08      0.05      0.00      0.20 1.00
 sd(InstMon06)        0.11      0.07      0.01      0.25 1.00
 sd(InstMon07)        0.11      0.06      0.01      0.24 1.00
 sd(InstMon08)        0.43      0.05      0.33      0.54 1.00
 sd(InstMon09)        0.23      0.06      0.11      0.37 1.00
 sd(InstMon10)        0.21      0.09      0.02      0.37 1.00
 sd(InstMon11)        0.19      0.10      0.01      0.41 1.00
 sd(InstMon12)        0.11      0.08      0.00      0.30 1.00
 sd(ProdMon02)        0.30      0.07      0.17      0.46 1.00
 sd(ProdMon03)        0.28      0.08      0.13      0.44 1.00
 sd(ProdMon04)        0.22      0.08      0.05      0.38 1.00
 sd(ProdMon05)        0.43      0.07      0.30      0.57 1.00
 sd(ProdMon06)        0.22      0.10      0.03      0.41 1.00
 sd(ProdMon07)        0.27      0.08      0.13      0.44 1.00
 sd(ProdMon08)        0.60      0.10      0.41      0.81 1.00
```

```
sd(ProdMon09)          0.49      0.09      0.32      0.68 1.00
sd(ProdMon10)          0.17      0.11      0.01      0.43 1.00
sd(ProdMon11)          0.29      0.14      0.03      0.57 1.00
sd(ProdMon12)          0.21      0.17      0.01      0.63 1.00

Population-Level Effects:
           Estimate Est.Error l-95% CI u-95% CI
Intercept     -4.38      0.12     -4.60     -4.14 1.00
Age>1         -0.87      0.05     -0.97     -0.77 1.00
InstMon02     -0.32      0.11     -0.53     -0.11 1.00
InstMon03     -0.66      0.10     -0.85     -0.47 1.00
InstMon04     -0.48      0.09     -0.65     -0.31 1.00
InstMon05     -0.54      0.08     -0.69     -0.38 1.00
InstMon06     -0.68      0.08     -0.84     -0.51 1.00
InstMon07     -0.88      0.08     -1.03     -0.72 1.00
InstMon08     -0.91      0.09     -1.08     -0.72 1.00
InstMon09     -0.57      0.08     -0.73     -0.40 1.00
InstMon10     -0.54      0.09     -0.71     -0.37 1.00
InstMon11     -0.57      0.09     -0.75     -0.39 1.00
InstMon12     -0.71      0.09     -0.88     -0.53 1.00
ProdMon02     -0.07      0.08     -0.23      0.10 1.00
ProdMon03     -0.21      0.08     -0.37     -0.05 1.00
ProdMon04     -0.30      0.08     -0.45     -0.16 1.00
ProdMon05     -0.14      0.09     -0.32      0.03 1.00
ProdMon06     -0.34      0.08     -0.50     -0.19 1.00
ProdMon07     -0.37      0.08     -0.53     -0.21 1.00
ProdMon08     -0.32      0.11     -0.55     -0.11 1.00
ProdMon09     -0.35      0.11     -0.56     -0.15 1.00
ProdMon10     -0.38      0.09     -0.57     -0.20 1.00
ProdMon11     -0.39      0.11     -0.61     -0.19 1.00
ProdMon12     -0.24      0.13     -0.51     -0.01 1.00
```

The chain diagnostics of the Poisson model are nearly excellent. The $\widehat{R}$ values suggest that chains are mixed. The samples are pretty efficient since there are no parameters with a n_{eff} ratio lower than 0.1, and nearly all of them are larger than 0.5.

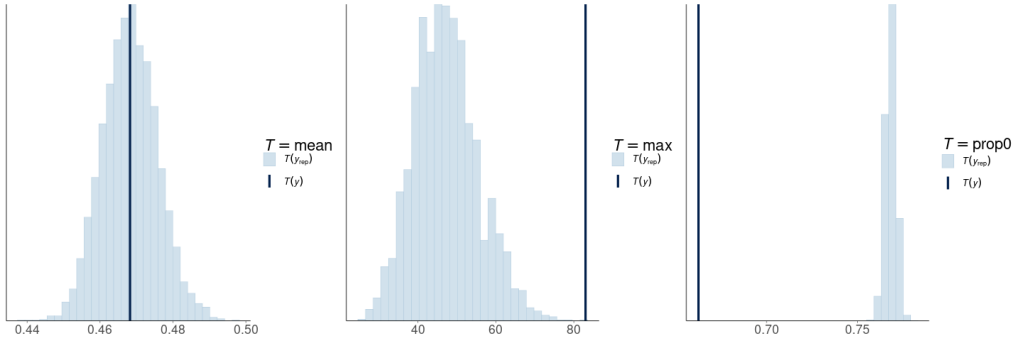
The posterior predictive plots of the Poisson model are presented in Figure 7.3. The mean posterior distribution is centered around the actual mean, but the model underestimates the maximum value and overestimates the zero portion in the data.

The posterior retrodictive plots are presented in Figure 7.4. For this problem, the maximum value of the response equals 83, and there is only one observation between y equals 40 and 83. Thus results are presented for two intervals of y, $[0, 5]$ and $[35, 40]$. Looking at Figure 7.4, one can deduce that model can not make an accurate estimate when there is only one defective refrigerator. The count of zeros in posterior retrodictive distribution is extremely higher than the zero counts in the data suggesting that most of the observations with one failed refrigerator are estimated as zero failure. Furthermore, Most of the values of y are not in the 99% posterior retrodictive distribution.



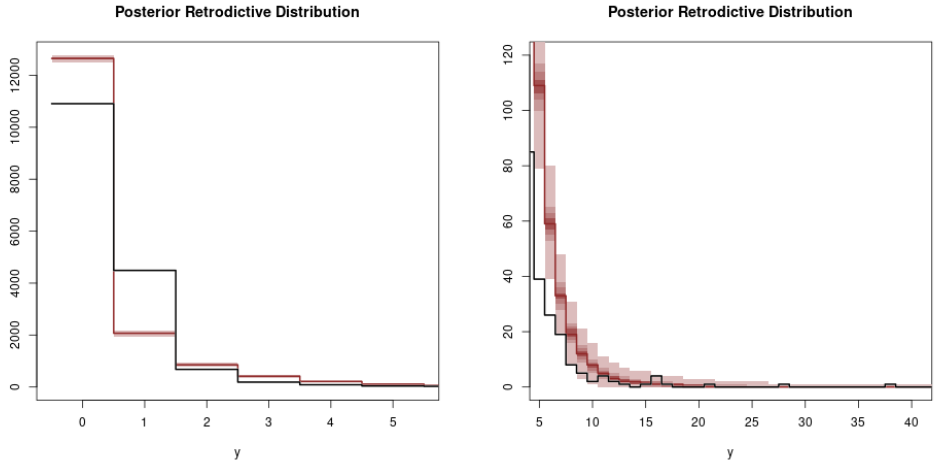
(a) $\widehat{R}$ values (b) Ratio of effective sample size

Figure 7.2: Chain diagnostics



(a) Mean (b) Max (c) Zero portion

Figure 7.3: Posterior predictive check on mean, max, and zero portion



(a) Interval [0,5] (b) Interval [5,40]

Figure 7.4: Posterior retrodictive check

Figure 7.5 shows the posterior prediction intervals for each installation month. Some intervals are quite close to the actual y values, but there are many over- and under-estimations in the prediction. Figure 7.6 shows the posterior prediction intervals for each production month. A similar interpretation can make for Figure 7.6.

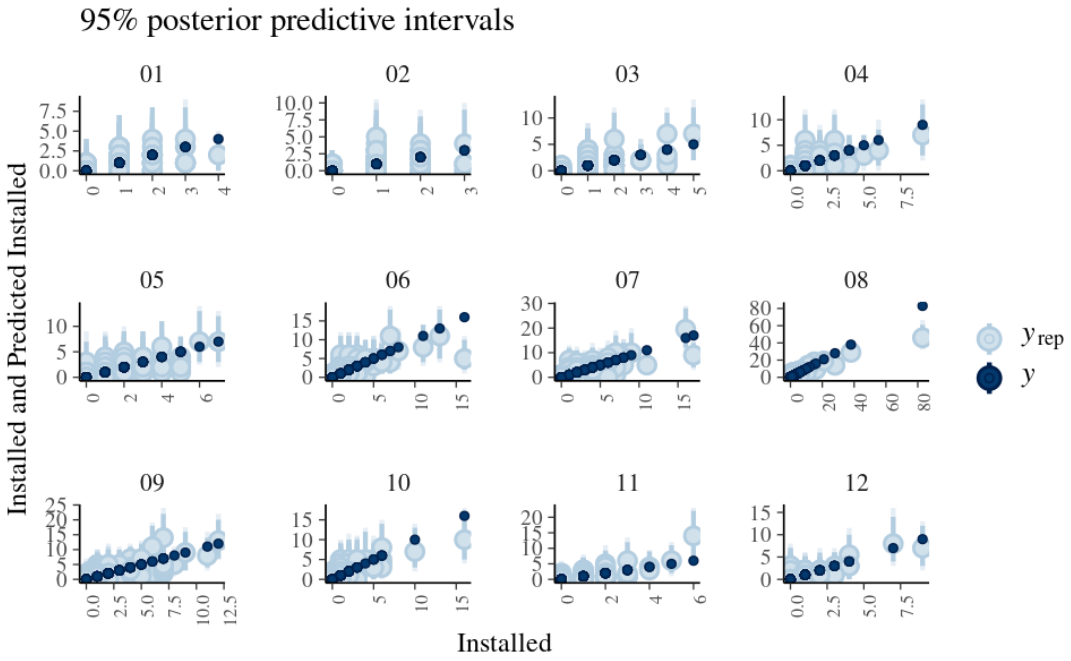


Figure 7.5: Posterior predictive distribution for each installation month

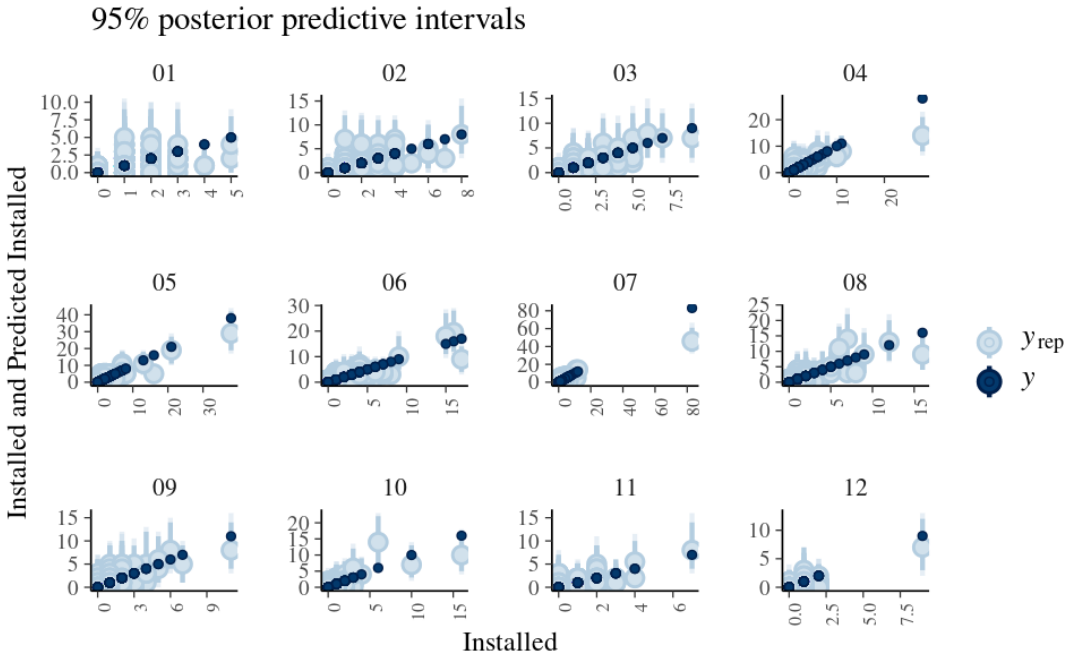


Figure 7.6: Posterior predictive distribution for each production month

7.2.2 Negative Binomial Model Diagnostics

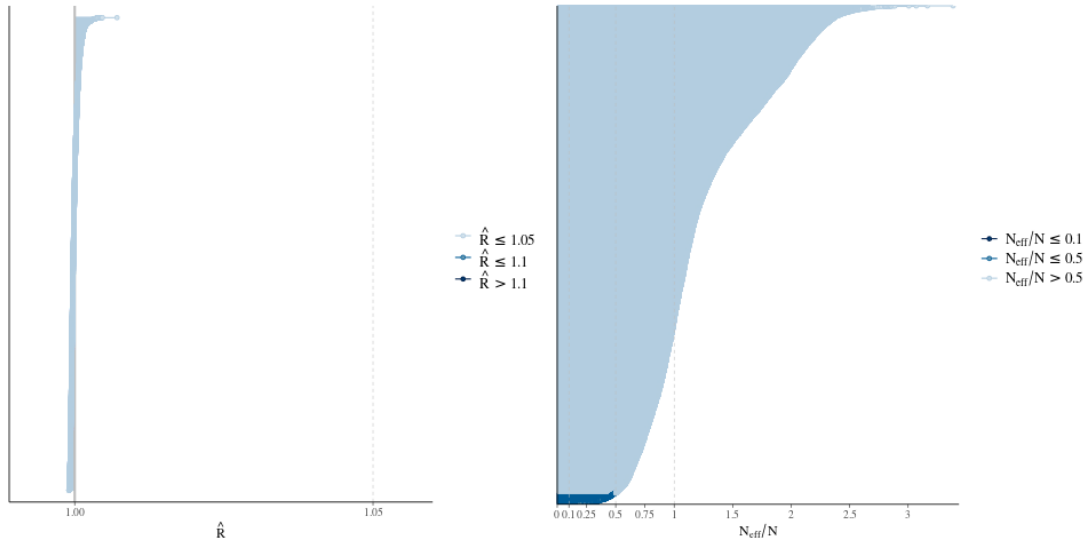
Listings 7.2 shows the summary of the negative binomial model. The estimated model parameters are nearly the same as the estimated parameters of the Poisson model. Furthermore, the inverse-overdispersion parameter is estimated as 152.56. This means that data is not overdispersed, and negative binomial distribution acts like Poisson distribution. The negative binomial model's diagnostic plots are the same as the Poisson model. Thus the same interpretations can be made.

Listing 7.2: Summary of negative binomial model

```
 Family: negbinomial
  Links: mu = log; shape = identity
Formula: Failed ~ 1 + AgeFailBinary+ InstMon + ProdMon
                + offset(log(Installed))
                + (1 + AgeFailBinary + InstMon
                + ProdMon | Model)
Data: FailureTrain (Number of observations: 16464)
Draws: 4 chains, each with iter = 1000; warmup = 0; thin = 1;
         total post-warmup draws = 4000

Group-Level Effects:
~Model (Number of levels: 141)
                Estimate Est.Error l-95% CI u-95% CI Rhat
sd(Intercept)       0.79      0.08      0.66      0.95 1.00
sd(Age>1)           0.33      0.04      0.25      0.42 1.00
sd(InstMon02)       0.29      0.11      0.07      0.50 1.00
sd(InstMon03)       0.18      0.09      0.01      0.37 1.00
sd(InstMon04)       0.08      0.06      0.00      0.24 1.00
sd(InstMon05)       0.07      0.05      0.00      0.19 1.00
sd(InstMon06)       0.11      0.07      0.01      0.25 1.00
sd(InstMon07)       0.11      0.06      0.01      0.23 1.00
sd(InstMon08)       0.43      0.06      0.33      0.54 1.00
sd(InstMon09)       0.23      0.07      0.11      0.37 1.00
sd(InstMon10)       0.20      0.09      0.02      0.37 1.00
```

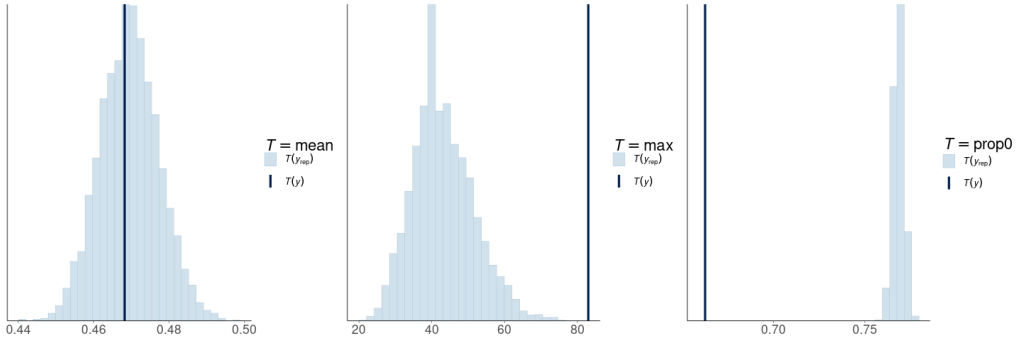
```
sd(InstMon11)     0.18      0.11     0.01     0.40 1.00
sd(InstMon12)     0.11      0.08     0.01     0.31 1.00
sd(ProdMon02)     0.30      0.08     0.16     0.47 1.00
sd(ProdMon03)     0.28      0.08     0.12     0.43 1.00
sd(ProdMon04)     0.21      0.08     0.04     0.37 1.00
sd(ProdMon05)     0.42      0.07     0.30     0.57 1.00
sd(ProdMon06)     0.21      0.10     0.02     0.39 1.00
sd(ProdMon07)     0.27      0.08     0.11     0.43 1.00
sd(ProdMon08)     0.60      0.10     0.41     0.82 1.00
sd(ProdMon09)     0.48      0.09     0.32     0.67 1.00
sd(ProdMon10)     0.18      0.12     0.01     0.45 1.00
sd(ProdMon11)     0.29      0.14     0.04     0.57 1.00
sd(ProdMon12)     0.21      0.16     0.01     0.60 1.00

Population-Level Effects:
             Estimate Est.Error l-95% CI u-95% CI Rhat
Intercept       -4.37      0.11    -4.59    -4.15 1.00
Age>1           -0.87      0.05    -0.97    -0.77 1.00
InstMon02       -0.32      0.11    -0.54    -0.11 1.00
InstMon03       -0.66      0.10    -0.84    -0.48 1.00
InstMon04       -0.48      0.09    -0.65    -0.31 1.00
InstMon05       -0.54      0.08    -0.69    -0.37 1.00
InstMon06       -0.68      0.08    -0.83    -0.52 1.00
InstMon07       -0.89      0.08    -1.05    -0.72 1.00
InstMon08       -0.91      0.09    -1.10    -0.73 1.00
InstMon09       -0.57      0.09    -0.73    -0.40 1.00
InstMon10       -0.54      0.09    -0.72    -0.37 1.00
InstMon11       -0.57      0.09    -0.75    -0.39 1.00
InstMon12       -0.71      0.09    -0.88    -0.53 1.00
ProdMon02       -0.07      0.08    -0.23     0.10 1.00
ProdMon03       -0.20      0.08    -0.37    -0.05 1.00
ProdMon04       -0.30      0.08    -0.45    -0.15 1.00
ProdMon05       -0.13      0.09    -0.31     0.04 1.00
ProdMon06       -0.34      0.08    -0.50    -0.19 1.00
ProdMon07       -0.37      0.08    -0.53    -0.21 1.00
```

```
ProdMon08    −0.32      0.11     −0.55     −0.10 1.00
ProdMon09    −0.35      0.10     −0.56     −0.15 1.00
ProdMon10    −0.38      0.09     −0.57     −0.20 1.00
ProdMon11    −0.39      0.11     −0.61     −0.18 1.00
ProdMon12    −0.24      0.13     −0.50     −0.01 1.00

Family Specific Parameters:
      Estimate Est.Error l−95% CI u−95% CI Rhat
shape   152.56     80.50    58.33   369.63 1.00
```



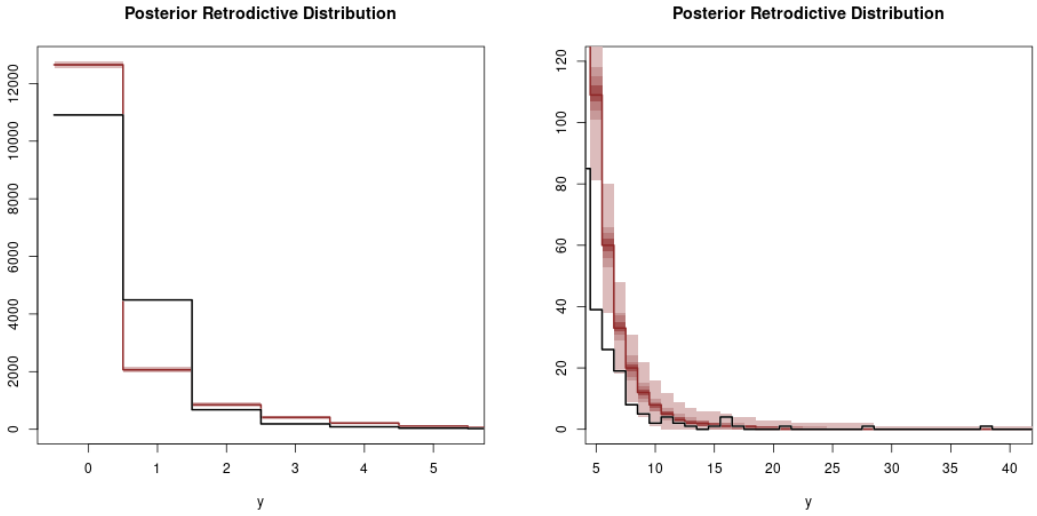
(a) $\widehat{R}$ values (b) Ratio of effective sample size

Figure 7.7: Chain diagnostics



(a) Mean (b) Max (c) Zero portion

Figure 7.8: Posterior predictive check on mean, max, and zero portion



(a) Interval [0,5] (b) Interval [5,40]

Figure 7.9: Posterior retrodictive check

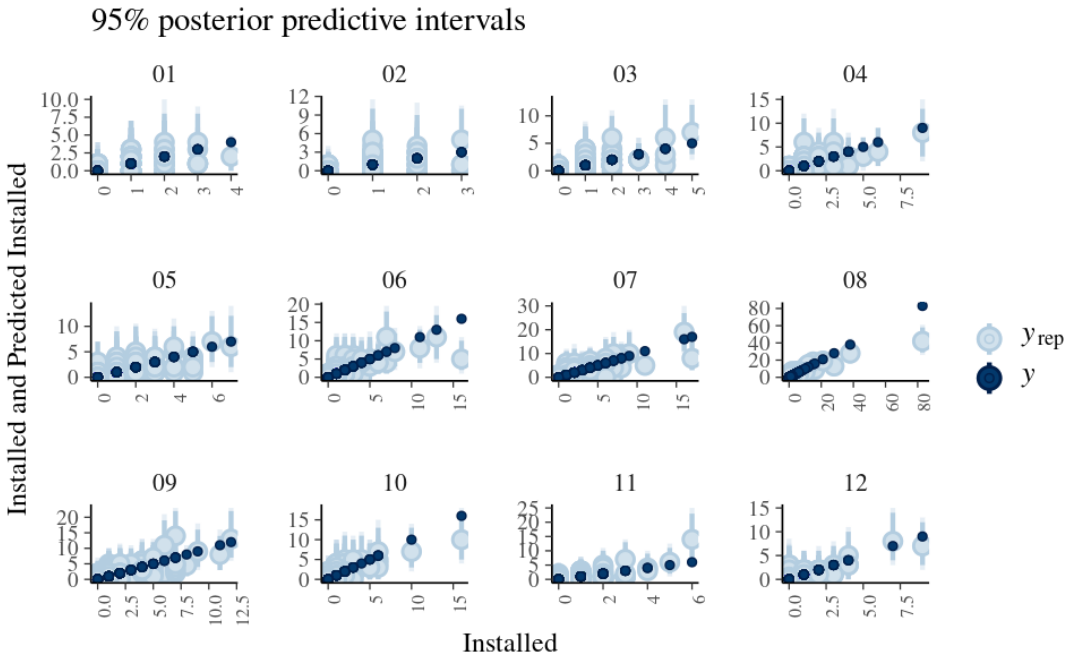


Figure 7.10: Posterior predictive distribution for each installation month

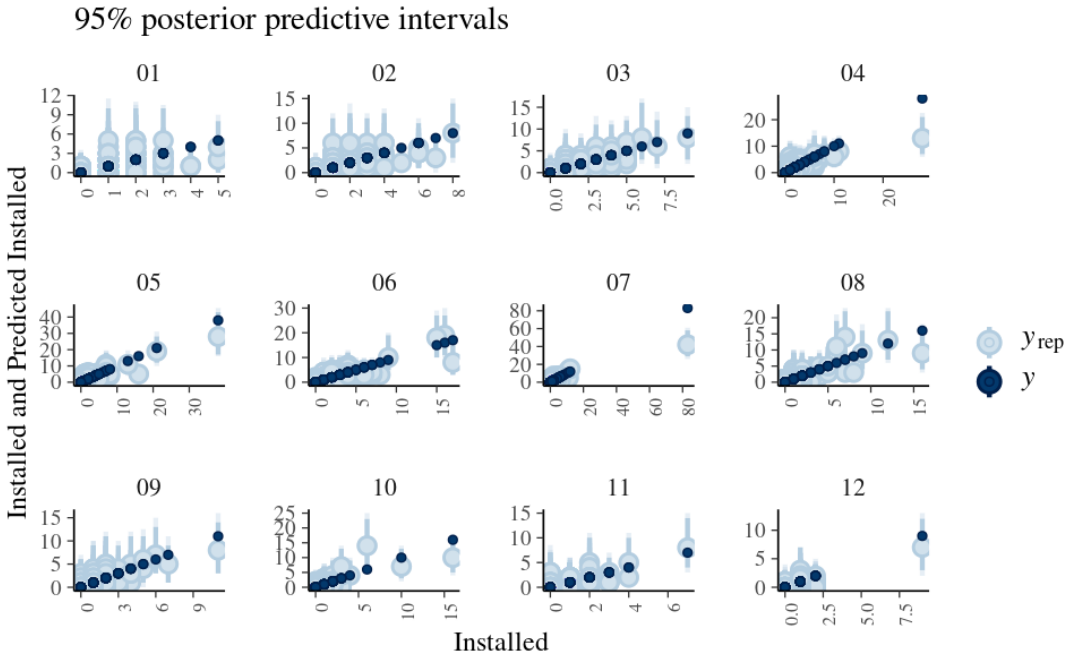


Figure 7.11: Posterior predictive distribution for each production month

7.2.3 Binomial Model Diagnostics

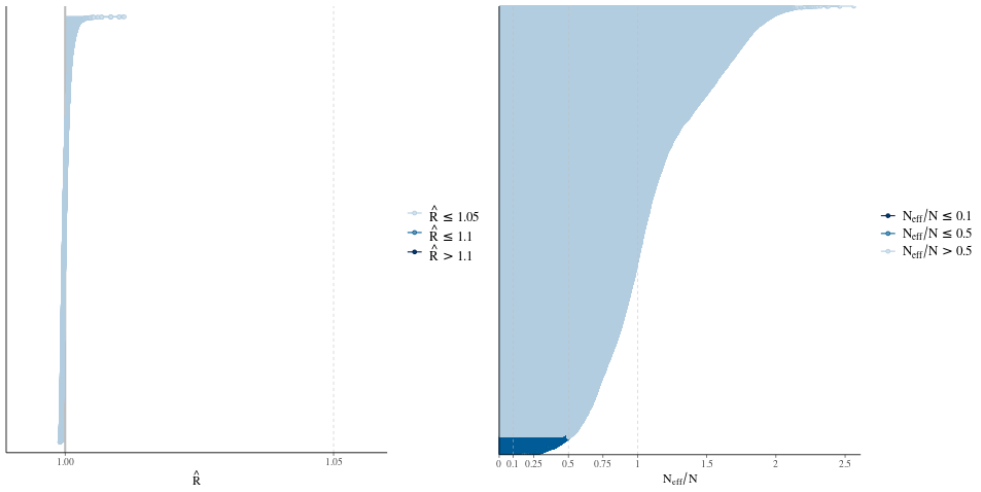
Listings 7.3 shows the summary of the binomial model. Estimated model parameters are close to the parameters of Poisson and negative binomial models. Furthermore, the binomial model's diagnostics plots are the same as previous diagnostic plots. This indicates that the three models are more or less the same, and their accuracy results should be close.

Listing 7.3: Summary of binomial model

```
 Family: binomial
  Links: mu = logit
Formula: Failed | trials(Installed) ~ 1 + AgeFailBinary + InstMon
                 + ProdMon + (1 + AgeFailBinary + InstMon
                 + ProdMon | Model)
Data: FailureTrain (Number of observations: 16464)
Draws: 4 chains, each with iter = 1000; warmup = 0; thin = 1;
         total post-warmup draws = 4000

Group-Level Effects:
~Model (Number of levels: 141)
               Estimate Est.Error l-95% CI u-95% CI Rhat
sd(Intercept)      0.81      0.08      0.67      0.97 1.00
sd(Age>1)          0.34      0.04      0.26      0.42 1.00
sd(InstMon02)      0.29      0.11      0.07      0.52 1.00
sd(InstMon03)      0.17      0.10      0.01      0.37 1.00
sd(InstMon04)      0.08      0.06      0.00      0.23 1.00
sd(InstMon05)      0.08      0.05      0.00      0.20 1.00
sd(InstMon06)      0.12      0.07      0.01      0.27 1.01
sd(InstMon07)      0.11      0.06      0.01      0.24 1.00
sd(InstMon08)      0.43      0.06      0.33      0.55 1.00
sd(InstMon09)      0.24      0.06      0.12      0.36 1.00
sd(InstMon10)      0.21      0.09      0.03      0.39 1.00
sd(InstMon11)      0.19      0.10      0.02      0.40 1.00
sd(InstMon12)      0.11      0.08      0.00      0.29 1.00
```

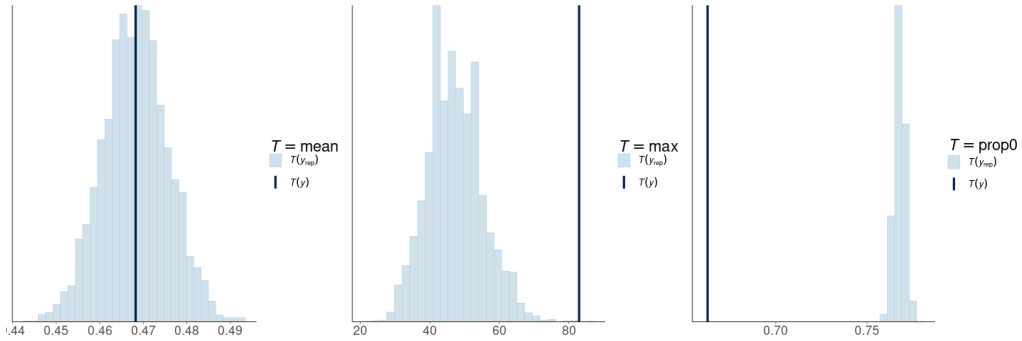
```
sd(ProdMon02)      0.30      0.07      0.17      0.45 1.00
sd(ProdMon03)      0.28      0.08      0.12      0.44 1.00
sd(ProdMon04)      0.21      0.08      0.04      0.37 1.00
sd(ProdMon05)      0.43      0.07      0.30      0.58 1.00
sd(ProdMon06)      0.22      0.09      0.03      0.41 1.00
sd(ProdMon07)      0.28      0.08      0.13      0.44 1.00
sd(ProdMon08)      0.60      0.10      0.41      0.81 1.00
sd(ProdMon09)      0.49      0.09      0.32      0.69 1.00
sd(ProdMon10)      0.18      0.12      0.01      0.44 1.00
sd(ProdMon11)      0.29      0.14      0.03      0.57 1.00
sd(ProdMon12)      0.21      0.16      0.01      0.62 1.00

Population-Level Effects:
           Estimate Est.Error l-95% CI u-95% CI Rhat
Intercept     -4.36      0.12     -4.58     -4.13 1.01
Age>1         -0.88      0.05     -0.98     -0.78 1.00
InstMon02     -0.33      0.11     -0.55     -0.11 1.00
InstMon03     -0.67      0.10     -0.85     -0.48 1.00
InstMon04     -0.49      0.09     -0.66     -0.31 1.00
InstMon05     -0.54      0.08     -0.70     -0.39 1.00
InstMon06     -0.68      0.08     -0.84     -0.53 1.00
InstMon07     -0.89      0.08     -1.06     -0.74 1.00
InstMon08     -0.92      0.09     -1.10     -0.74 1.00
InstMon09     -0.57      0.09     -0.75     -0.41 1.00
InstMon10     -0.55      0.09     -0.73     -0.38 1.00
InstMon11     -0.58      0.09     -0.76     -0.40 1.00
InstMon12     -0.71      0.09     -0.89     -0.54 1.00
ProdMon02     -0.07      0.08     -0.24      0.09 1.00
ProdMon03     -0.21      0.08     -0.36     -0.05 1.00
ProdMon04     -0.31      0.08     -0.46     -0.16 1.00
ProdMon05     -0.14      0.09     -0.31      0.04 1.00
ProdMon06     -0.34      0.08     -0.51     -0.19 1.00
ProdMon07     -0.37      0.08     -0.53     -0.22 1.00
ProdMon08     -0.32      0.12     -0.56     -0.10 1.00
ProdMon09     -0.36      0.10     -0.57     -0.16 1.00
```

62	ProdMon10	−0.38	0.10	−0.58	−0.20	1.00
63	ProdMon11	−0.39	0.11	−0.60	−0.18	1.00
64	ProdMon12	−0.25	0.13	−0.51	−0.01	1.00



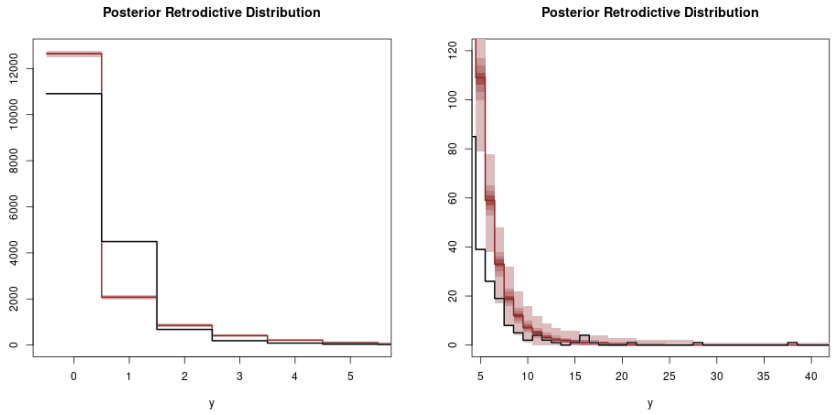
(a) $\widehat{R}$ values (b) Ratio of effective sample size

Figure 7.12: Chain diagnostics



(a) Mean (b) Max (c) Zero portion

Figure 7.13: Posterior predictive check on mean, max, and zero portion



(a) Interval [0,5] (b) Interval [5,40]

Figure 7.14: Posterior retrodictive check

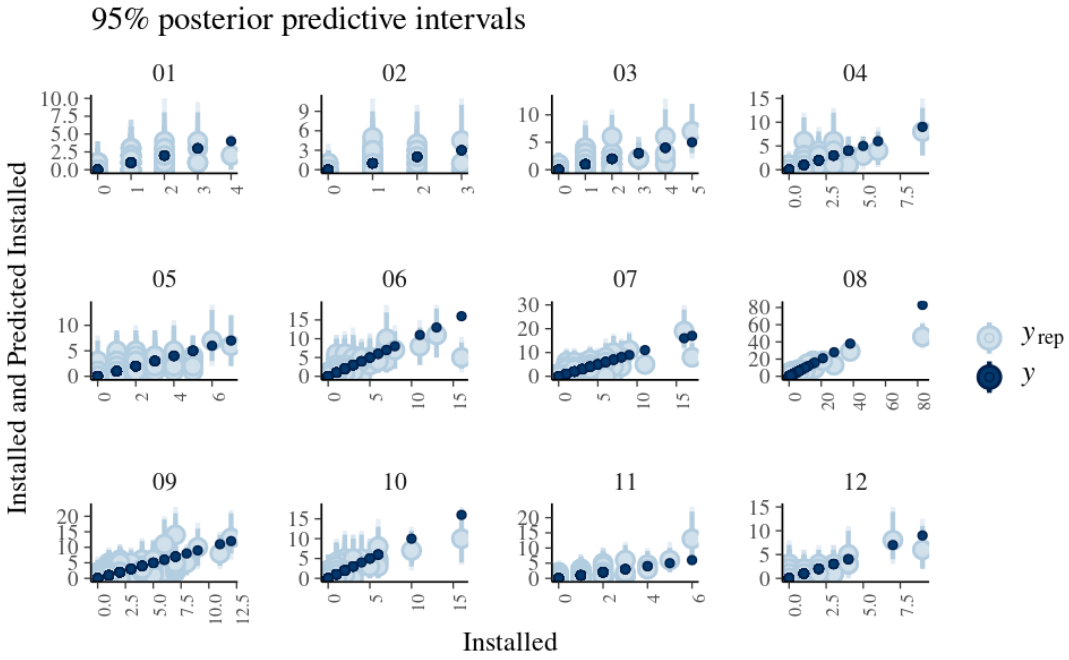


Figure 7.15: Posterior predictive distribution for each installation month

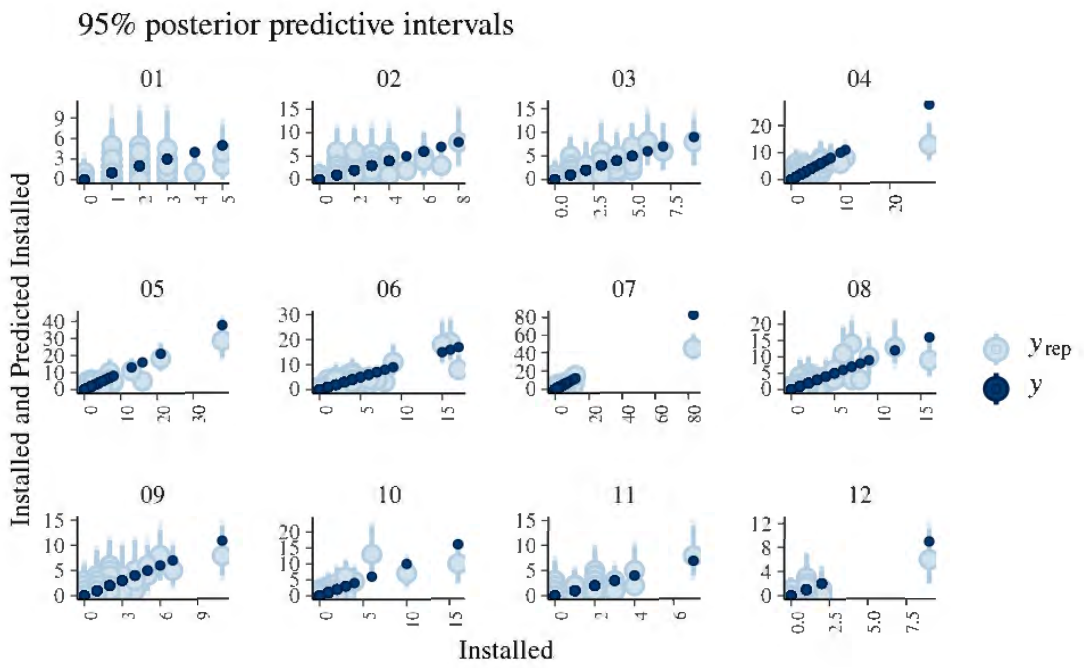


Figure 7.16: Posterior predictive distribution for each production month

As discussed before, the binomial distribution is asymptotically equal to the Poisson distribution; thus, obtaining similar diagnostics plots might be expected. On the other hand, obtaining similar results from the Poisson and negative binomial model indicates the data is not overdispersed.The negative binomial distribution acts like the Poisson distribution when the dispersion parameter is close to zero; this is a suitable explanation for our case since the mean of the posterior distribution of the dispersion parameter equals 0.006.

7.2.4 Beta-Binomial Model Diagnostics

Listings 7.4 shows the summary of beta-binomial model. Estimated parameters close to other models' parameters. Inverse-overdispersion parameter is denoted as phi in the Listings 7.4, and its estimation is quite high.

Listing 7.4: Summary of beta-binomial model

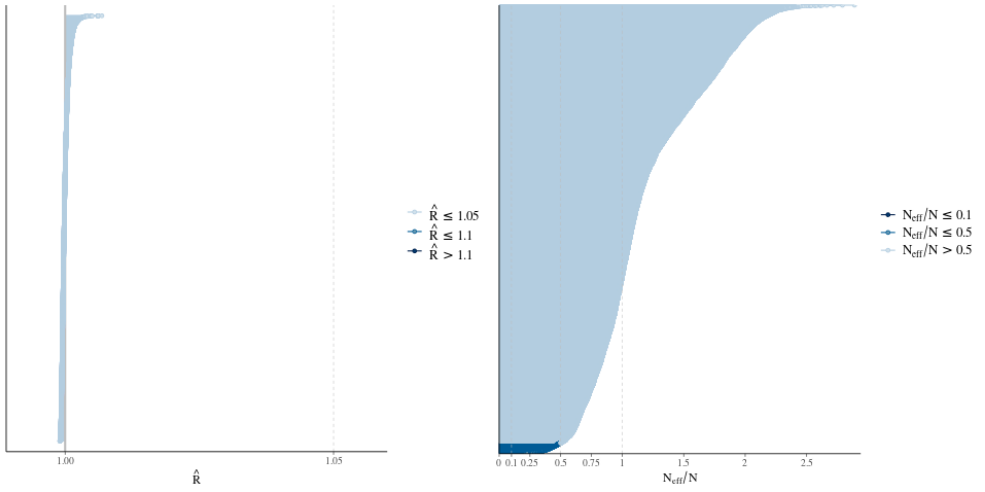
```
2  Family: beta_binomial
3  Links: mu = logit; phi = identity
4  Formula: Failed | trials(Installed) ~ 1 + AgeFailBinary
5                  + InstMon + ProdMon + (1 + AgeFailBinary
6                  + InstMon + ProdMon | Model)
7  Data: FailureTrain (Number of observations: 16464)
8  Draws: 4 chains, each with iter = 1000; warmup = 0; thin = 1;
9           total post-warmup draws = 4000

11 Group-Level Effects:
12 ~Model (Number of levels: 141)
13                  Estimate Est.Error l-95% CI u-95% CI Rhat
14 sd(Intercept)        0.79      0.07      0.66      0.95 1.00
15 sd(Age>1)            0.31      0.04      0.23      0.40 1.00
16 sd(InstMon02)        0.28      0.11      0.05      0.51 1.00
17 sd(InstMon03)        0.16      0.09      0.01      0.36 1.00
18 sd(InstMon04)        0.08      0.06      0.00      0.23 1.00
19 sd(InstMon05)        0.07      0.05      0.00      0.19 1.00
20 sd(InstMon06)        0.10      0.06      0.00      0.23 1.00
21 sd(InstMon07)        0.09      0.06      0.01      0.22 1.00
22 sd(InstMon08)        0.40      0.06      0.30      0.52 1.00
23 sd(InstMon09)        0.22      0.07      0.08      0.35 1.00
24 sd(InstMon10)        0.18      0.09      0.01      0.35 1.00
25 sd(InstMon11)        0.16      0.10      0.01      0.37 1.00
26 sd(InstMon12)        0.10      0.08      0.00      0.29 1.00
27 sd(ProdMon02)        0.29      0.07      0.17      0.45 1.00
28 sd(ProdMon03)        0.26      0.08      0.12      0.41 1.00
```

sd(ProdMon04)	0.16	0.09	0.01	0.34	1.00
sd(ProdMon05)	0.41	0.07	0.29	0.55	1.00
sd(ProdMon06)	0.20	0.10	0.02	0.39	1.00
sd(ProdMon07)	0.25	0.08	0.09	0.41	1.00
sd(ProdMon08)	0.58	0.10	0.39	0.80	1.00
sd(ProdMon09)	0.47	0.09	0.30	0.66	1.00
sd(ProdMon10)	0.16	0.11	0.01	0.42	1.01
sd(ProdMon11)	0.27	0.13	0.03	0.54	1.00
sd(ProdMon12)	0.20	0.16	0.01	0.58	1.00

Population–Level Effects:

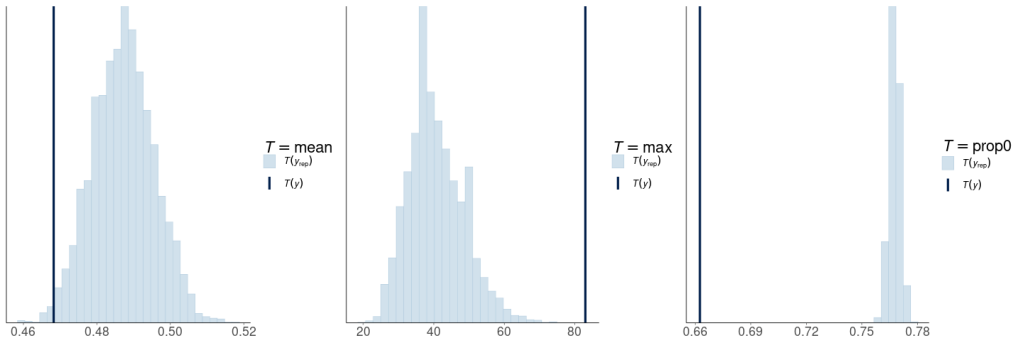
	Estimate	Est.Error	l–95% CI	u–95% CI	Rhat
Intercept	−4.38	0.12	−4.61	−4.15	1.00
Age>1	−0.85	0.05	−0.95	−0.75	1.00
InstMon02	−0.31	0.11	−0.53	−0.10	1.00
InstMon03	−0.65	0.10	−0.83	−0.46	1.00
InstMon04	−0.47	0.09	−0.65	−0.30	1.00
InstMon05	−0.52	0.08	−0.69	−0.36	1.00
InstMon06	−0.66	0.08	−0.82	−0.50	1.00
InstMon07	−0.86	0.08	−1.02	−0.71	1.00
InstMon08	−0.88	0.09	−1.06	−0.71	1.00
InstMon09	−0.56	0.08	−0.73	−0.39	1.00
InstMon10	−0.53	0.09	−0.70	−0.36	1.00
InstMon11	−0.56	0.09	−0.74	−0.38	1.00
InstMon12	−0.69	0.09	−0.87	−0.51	1.00
ProdMon02	−0.08	0.08	−0.24	0.08	1.00
ProdMon03	−0.19	0.08	−0.34	−0.05	1.00
ProdMon04	−0.31	0.07	−0.46	−0.16	1.00
ProdMon05	−0.13	0.09	−0.30	0.05	1.00
ProdMon06	−0.33	0.08	−0.49	−0.17	1.00
ProdMon07	−0.35	0.08	−0.50	−0.20	1.00
ProdMon08	−0.30	0.11	−0.52	−0.09	1.00
ProdMon09	−0.34	0.10	−0.54	−0.14	1.00
ProdMon10	−0.37	0.09	−0.56	−0.19	1.00
ProdMon11	−0.38	0.11	−0.59	−0.18	1.00

```
64 ProdMon12          −0.25       0.12      −0.50      −0.01  1.00

66 Family Specific Parameters:
67      Estimate Est.Error l−95% CI u−95% CI Rhat Bulk_ESS Tail_ESS
68 phi  7033.49     581.60  5986.41  8251.19 1.00
```



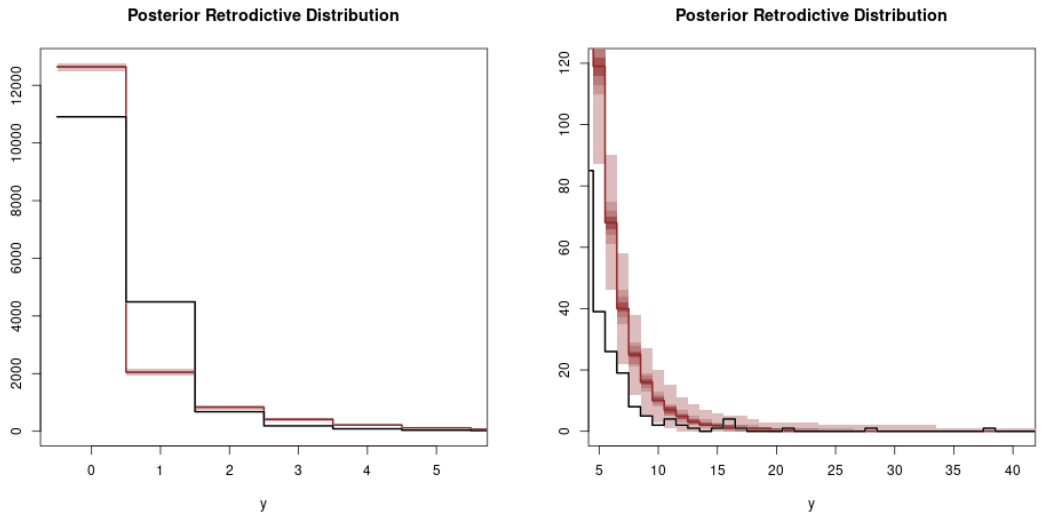
(a) $\widehat{R}$ values (b) Ratio of effective sample size

Figure 7.17: Chain diagnostics



(a) Mean (b) Max (c) Zero portion

Figure 7.18: Posterior predictive check on mean, max, and zero portion



(a) Interval [0,5] (b) Interval [5,40]

Figure 7.19: Posterior retrodictive check

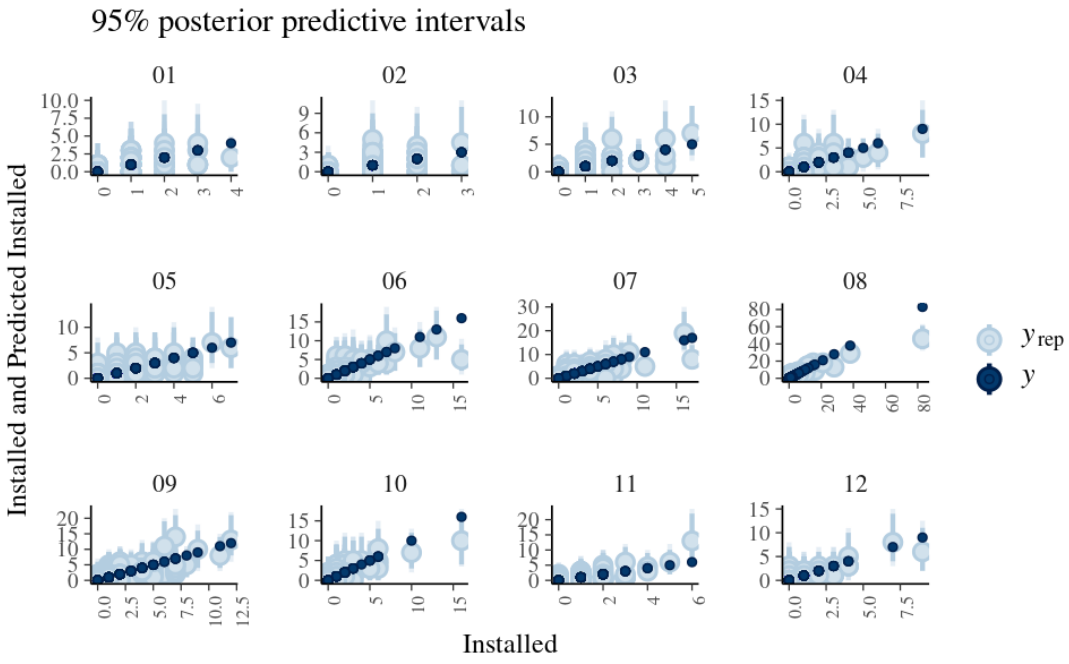


Figure 7.20: Posterior predictive distribution for each installation month

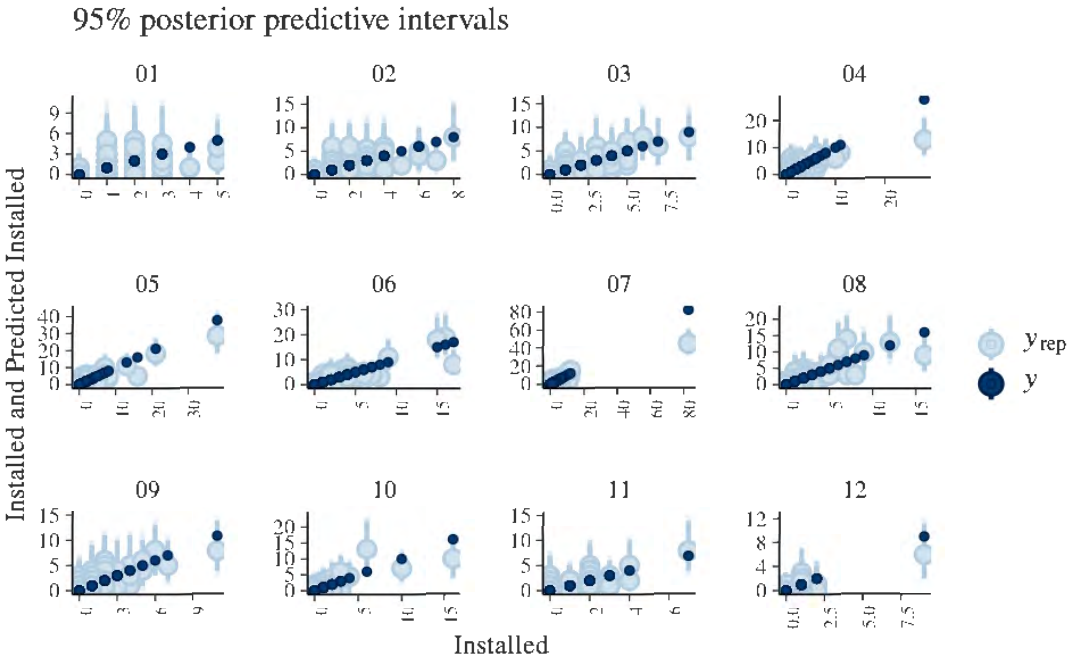


Figure 7.21: Posterior predictive distribution for each production month

Chain diagnostic of the beta-binomial model suggests that both mixing and effectiveness of the samples are pretty good. Unlike other models, the beta-binomial model overestimates the mean of the data. Figure 7.19b shows that the posterior retrodictive distribution of the model is slightly worse than other models'. Both Figures 7.20 and 7.21 indicates that there are many inaccurate predictions.

7.2.5 Poisson Hurdle Model Diagnostics

Listing 7.5 shows the summary of the Poisson hurdle model. By looking at the population-level effects, one can deduce that refrigerators older than one month and installed in December have a higher zero failure probability. Group-level effect of the intercept of the zero failure probability is strong. This means that refrigerator models influence the age effect and installation month effect of

January.

Listing 7.5: Summary of Poisson hurdle model

```
 Family: ~ hurdle_poisson
 Links: mu = log; zero = logit
 Formula: Failed ~ 1 + AgeFailBinary + offset(log(Installed))
                     + InstMon + ProdMon + (1 + InstMon
                     + AgeFailBinary + ProdMon | Model)
            ZeroFailed ~ 1 + AgeFailBinary + InstMon
                     + (1 + AgeFailBinary + InstMon | Model)
 Data: FailureTrain (Number of observations: 16464)
 Draws: 4 chains, each with iter = 1000; warmup = 0; thin = 1;
          total post-warmup draws = 4000

 Group-Level Effects:
 ~Model (Number of levels: 141)
                 Estimate Est.Error l-95% CI u-95% CI Rhat
 sd(Intercept)       0.71      0.09      0.55      0.89 1.00
 sd(InstMon02)       0.52      0.40      0.02      1.49 1.00
 sd(InstMon03)       0.21      0.17      0.01      0.61 1.00
 sd(InstMon04)       0.21      0.15      0.01      0.56 1.00
 sd(InstMon05)       0.13      0.10      0.00      0.36 1.00
 sd(InstMon06)       0.34      0.14      0.06      0.63 1.00
 sd(InstMon07)       0.20      0.11      0.02      0.44 1.00
 sd(InstMon08)       0.40      0.10      0.21      0.61 1.00
 sd(InstMon09)       0.12      0.10      0.00      0.37 1.00
 sd(InstMon10)       0.42      0.19      0.06      0.82 1.00
 sd(InstMon11)       0.31      0.20      0.02      0.77 1.00
 sd(InstMon12)       0.24      0.19      0.01      0.71 1.00
 sd(Age>1)           0.70      0.11      0.51      0.93 1.00
 sd(ProdMon02)       0.50      0.18      0.16      0.88 1.00
 sd(ProdMon03)       0.28      0.16      0.02      0.62 1.00
 sd(ProdMon04)       0.39      0.13      0.15      0.67 1.00
 sd(ProdMon05)       0.63      0.13      0.40      0.92 1.00
 sd(ProdMon06)       0.24      0.15      0.01      0.58 1.00
```

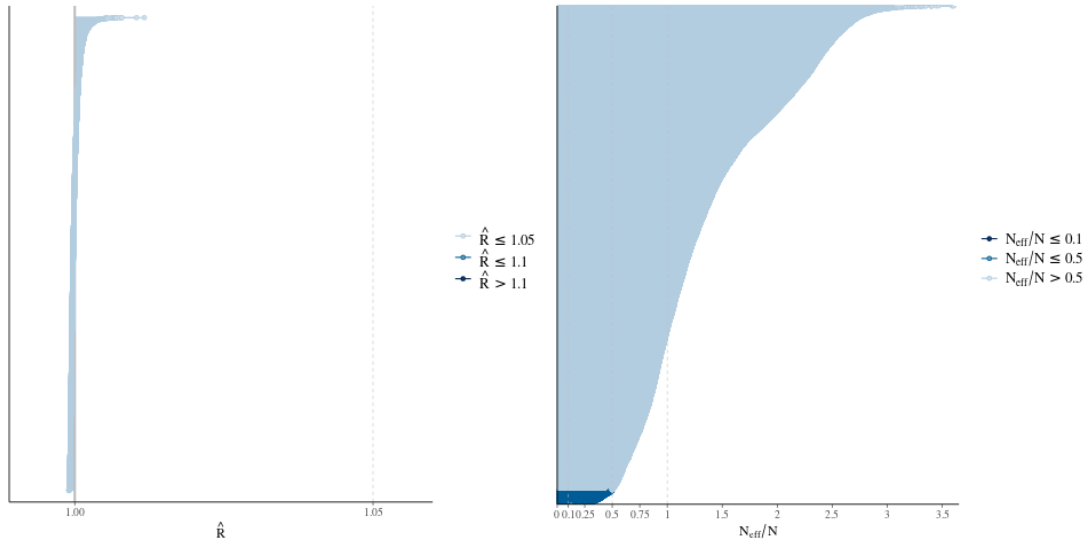
```
sd(ProdMon07)          0.36      0.13      0.13      0.63 1.00
sd(ProdMon08)          0.73      0.20      0.37      1.15 1.00
sd(ProdMon09)          0.32      0.18      0.02      0.70 1.00
sd(ProdMon10)          0.22      0.19      0.01      0.70 1.00
sd(ProdMon11)          0.26      0.19      0.01      0.72 1.00
sd(ProdMon12)          0.77      0.64      0.03      2.39 1.00
sd(zero_Intercept)     1.77      0.16      1.49      2.10 1.00
sd(zero_Age>1)         0.36      0.12      0.09      0.59 1.01
sd(zero_InstMon02)     0.39      0.19      0.03      0.76 1.00
sd(zero_InstMon03)     0.73      0.17      0.42      1.07 1.00
sd(zero_InstMon04)     0.19      0.13      0.01      0.49 1.00
sd(zero_InstMon05)     0.28      0.15      0.02      0.58 1.00
sd(zero_InstMon06)     0.39      0.14      0.10      0.67 1.00
sd(zero_InstMon07)     0.42      0.13      0.17      0.68 1.00
sd(zero_InstMon08)     0.63      0.11      0.43      0.86 1.00
sd(zero_InstMon09)     0.62      0.11      0.41      0.85 1.00
sd(zero_InstMon10)     0.51      0.13      0.26      0.76 1.00
sd(zero_InstMon11)     0.32      0.16      0.03      0.63 1.00
sd(zero_InstMon12)     0.85      0.16      0.56      1.17 1.01

Population-Level Effects:
                   Estimate Est.Error l-95% CI u-95% CI Rhat
Intercept            -5.82      0.24     -6.31     -5.38 1.00
zero_Intercept        0.20      0.20     -0.19      0.59 1.00
Age>1                -1.71      0.12     -1.94     -1.48 1.00
InstMon02            -0.69      0.37     -1.53     -0.04 1.00
InstMon03            -0.53      0.25     -1.03     -0.03 1.00
InstMon04            -0.32      0.24     -0.77      0.15 1.00
InstMon05            -0.08      0.21     -0.48      0.35 1.00
InstMon06            -0.30      0.22     -0.71      0.15 1.00
InstMon07            -0.38      0.20     -0.78      0.02 1.00
InstMon08            -0.51      0.22     -0.90     -0.07 1.00
InstMon09            -0.04      0.21     -0.42      0.39 1.00
InstMon10            -0.37      0.24     -0.85      0.11 1.00
InstMon11            -0.58      0.24     -1.05     -0.10 1.00
```

InstMon12	−0.40	0.25	−0.91	0.08	1.00
ProdMon02	0.13	0.19	−0.25	0.49	1.00
ProdMon03	−0.16	0.17	−0.52	0.16	1.00
ProdMon04	−0.16	0.17	−0.51	0.18	1.00
ProdMon05	0.03	0.19	−0.33	0.39	1.00
ProdMon06	−0.03	0.17	−0.38	0.29	1.00
ProdMon07	−0.09	0.17	−0.42	0.24	1.00
ProdMon08	−0.26	0.22	−0.71	0.17	1.00
ProdMon09	−0.22	0.19	−0.59	0.15	1.00
ProdMon10	−0.09	0.21	−0.51	0.30	1.00
ProdMon11	−0.20	0.22	−0.64	0.24	1.00
ProdMon12	−0.62	0.59	−2.13	0.23	1.00
zero_Age>1	2.16	0.09	1.98	2.34	1.00
zero_InstMon02	0.22	0.16	−0.09	0.56	1.00
zero_InstMon03	0.11	0.19	−0.26	0.49	1.00
zero_InstMon04	−0.32	0.13	−0.57	−0.07	1.00
zero_InstMon05	−0.59	0.13	−0.85	−0.34	1.00
zero_InstMon06	−0.84	0.13	−1.09	−0.57	1.00
zero_InstMon07	−0.98	0.13	−1.23	−0.72	1.00
zero_InstMon08	−1.07	0.14	−1.35	−0.78	1.00
zero_InstMon09	−0.51	0.15	−0.81	−0.21	1.00
zero_InstMon10	−0.06	0.15	−0.34	0.23	1.00
zero_InstMon11	0.46	0.14	0.19	0.75	1.00
zero_InstMon12	1.54	0.21	1.14	1.95	1.00

As expected, the zero portion of the posterior predictive distribution of the hurdle model is perfect, but obviously, it affects the mean of the response. Hence, the plot in Figure 7.23a is worse compared to its equivalents in other models. Also, the plot in Figure 7.23b indicates a red flag since there are some unusually large values.

However, the retrodictive plots in Figure 7.24 portray a better picture in terms of estimating frequencies of most of the values of y. The plot in Figure 7.24a suggests that the model can estimate the frequency of ones more accurately.

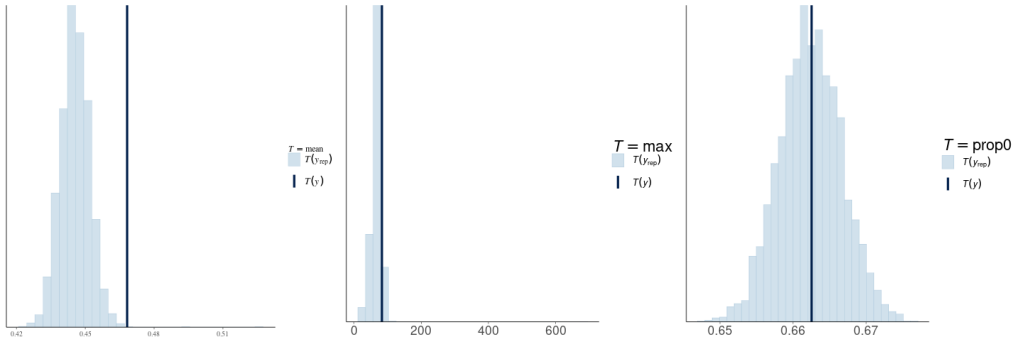
Furthermore, the amount of overestimation for the larger values of y decreased substantially.



(a) $\widehat{R}$ values (b) Ratio of effective sample size

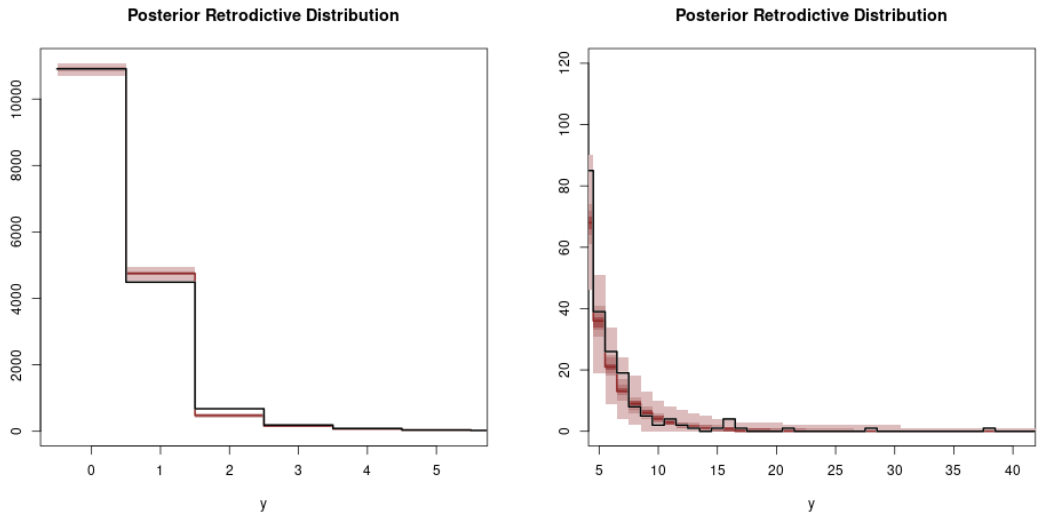
Figure 7.22: Chain diagnostics

Figures 7.25 and 7.26 indicate that some of the observations have a unreasonably large 95% posterior predictive interval. The estimations for those observations are unstable and distort the plot in Figure 7.23b.



(a) Mean (b) Max (c) Zero portion

Figure 7.23: Posterior predictive check on mean, max, and zero portion



(a) Interval [0,5] (b) Interval [5,40]

Figure 7.24: Posterior retrodictive check

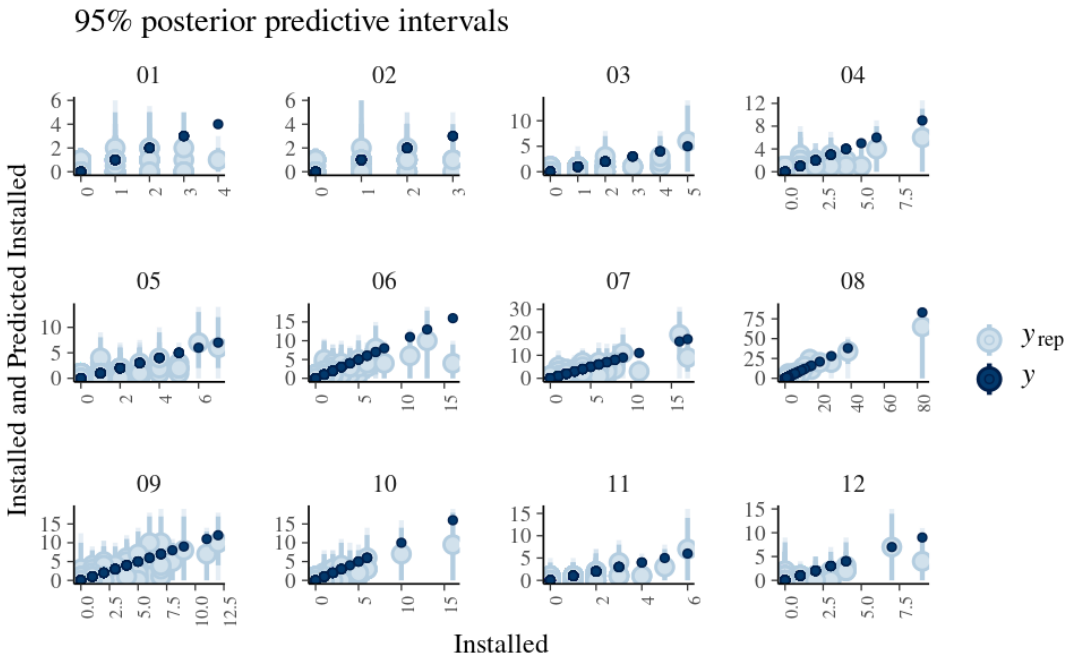


Figure 7.25: Posterior predictive distribution for each installation month

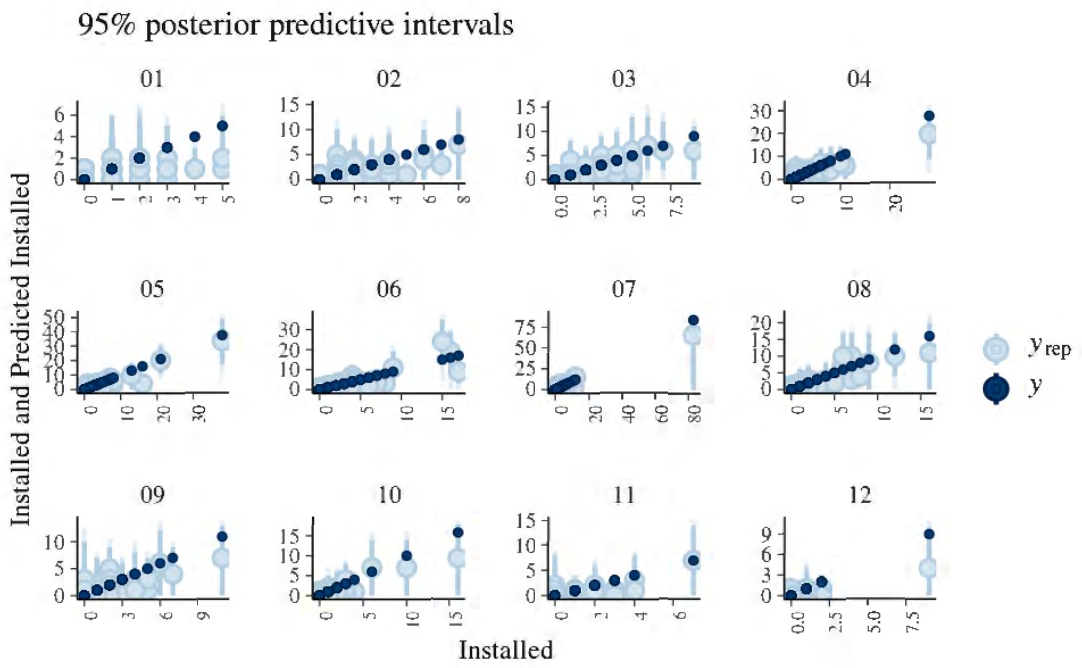


Figure 7.26: Posterior predictive distribution for each production month

Here we presented the explicit form of the selected model for the prediction of the number of in-service refrigerator failures. We decided to use the binomial model for estimating failure numbers. A detailed discussion on selecting final models is made in Chapter 8. In the explicit form of the model, we denote the grouping factor, namely the refrigerator and compressor model, with m. The explicit form of the model is presented in Equation 7.6

$$
\begin{aligned}
y^m_{\text{failed}} \mid y^m_{\text{failed}} > 0 &\sim \text{Binom}\left(n^m_{\text{installed}}, \mu^m\right), \\
\text{logit}\left(\mu^m\right) &= \beta_0 + \beta^m_1 \text{InstMon2} + \beta^m_2 \text{InstMon3} + \cdots + \beta^m_{11} \text{InstMon12} \\
&\quad + \beta^m_{12} \text{ProdMon2} + \beta^m_{13} \text{ProdMon3} + \cdots + \beta^m_{22} \text{ProdMon12} \\
&\quad + \beta^m_{23} \text{AgeFailBinary}, \\
\begin{bmatrix} \beta^m_0 \\ \beta^m_1 \\ \vdots \\ \beta^m_{23} \end{bmatrix} &\sim \mathcal{N}\left(\begin{bmatrix} \bar{\beta}_0 \\ \bar{\beta}_1 \\ \vdots \\ \bar{\beta}_{23} \end{bmatrix}, \Sigma \right),
\end{aligned}
$$

$$\Sigma = \begin{bmatrix} \sigma_0 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_{23} \end{bmatrix} L \begin{bmatrix} \sigma_0 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_{23} \end{bmatrix}, \tag{7.6}$$

$$\bar{\beta}_0, \bar{\beta}_1, \ldots, \bar{\beta}_{22} \overset{\text{iid}}{\sim} t_3\left(0, 10\right),$$

$$\sigma_0, \sigma, \ldots, \sigma_{23}, \overset{\text{iid}}{\sim} C^+\left(0, 3\right),$$

$$L \sim \text{LKJCorr}\left(2\right).$$

Chapter 8

Model Comparisons

In this chapter, the results of the models and their comparisons are presented. The proposed hierarchical models are compared with their equivalents in the frequentist approach. Since hierarchical models use the refrigerator model as the grouping factor, interaction effects of refrigerator models with other covariates are added to these models. To understand whether or not the hierarchical structure is needed, we also fitted non-hierarchical Bayesian models and their frequentist approach versions. These models do not include interaction effects, and the main effect of the refrigerator model is added to these models. Data sets are divided into training and test datasets of equal sizes, and models are trained and tested using them. The comparison is performed using the estimated loo, WAIC, root mean squared error (RMSE), normalized RMSE (normalized by average and normalized by standard deviation), and mean absolute error (MAE).

8.1 Comparisons of Installation Models

This section discusses the accuracy results of models that are fitted to predict the number of installed refrigerators. Table 8.1 presents the estimated loo and WAIC values of the Bayesian models. Most of the model's portion of the problematic observations is less than 0.5%. These observations have an estimated shape parameter of Pareto distribution larger than 0.7. The percentage of alarming observations for hierarchical Poisson and binomial models is larger than 20%. Similarly, the percentage of alarming observations for non-hierarchical Poisson and binomial models is approximately 4.5%. Thus their estimated loo and WAIC scores are not reliable. For other models, these scores are reliable; thus, a comparison is possible for only them. Generally, hierarchical models perform better according to these metrics. One can deduce that negative binomial and beta-binomial models yield comparable predictive accuracies. The hierarchical negative binomial hurdle model performs best. However, its non-hierarchical version gives the poorest result among all. This indicates that data gets too sparse when all the covariates are included and zeros are modeled separately.

Table 8.1: Bayesian Comparison Statistics for Models of Number of Installations

Model	Loo	WAIC
Hierarchical Poisson	542568	601702
Hierarchical Negative Binomial	119450	119418
Hierarchical Binomial	618246	697307
Hierarchical Beta-Binomial	119544	119520
Hierarchical Hurdle	**118542**	**118474**
Non-Hierarchical Poisson	826121	845366
Non-Hierarchical Negative Binomial	120138	120133
Non-Hierarchical Binomial	955340	980483
Non-Hierarchical Beta-Binomial	120338	120336
Non-Hierarchical Hurdle	122783	122777

For obtaining a much-stabilized comparison, K-fold cross-validation should be used. However, that kind of intensive application sometimes is infeasible. Considering the run time of fitting a Bayesian model, it is not possible for us to compare models using K-fold CV. Hence we could only calculate the training and test accuracies of the models. Table 8.2 shows the training MAE, RMSE, ARMSE (normalized with average), and SRMSE (normalized with standard deviation) values of models. The results suggest that the main effect models fitted using MLE are similar to non-hierarchical Bayesian models. Comparing these results with test accuracies in Table 8.3, one can deduce that main effect models are not overfitting. Furthermore, this also indicates that for the main effect models, data is informative enough and dominates the prior information; thus, using MCMC doesn't have an advantage over the MLE. On the contrary, the Poisson and negative binomial models with interaction effect and fitted by MLE are clearly overfitting. Although their training errors are good enough, test errors are unreasonably large. This observation is not valid for the interaction effect binomial model since its test results are acceptable. There is no solution for the negative binomial hurdle model with interaction effects due to sparsity in the data; hence it's not included in Table 8.2 and 8.3.

Most test accuracies are comparable, with few exceptions. Extremely high test errors for interaction models indicate using a hierarchical structure is highly beneficial when the interaction effects are needed. In other words, no-pooling interaction effect models are overfitting when the data is sparse, but partial-pooling models work well. On the other hand, hierarchical models can not outperform the main effects models. Although hierarchical negative binomial and beta-binomial models have the lowest MAE, main effect models perform better in squared error based metrics. One can deduce that the main effect models are slightly better in estimating larger count values.

Table 8.2: Comparison Statistics on Train Data
(NB=negative binomial, H=hierarchical, BB=beta-binomial, NH=non-hierarchical)

Model	MAE	RMSE	ARMSE	SRMSE
Poisson (MLE)	49.73	176.02	2.01	0.55
NB (MLE)	53.14	216.83	2.48	0.68
Binomial (MLE)	49.66	176.12	2.02	0.55
Hurdle (MLE)	53.42	218.10	2.50	0.68
NH Poisson (MCMC)	49.73	176.02	2.01	0.55
NH NB (MCMC)	53.19	216.76	2.48	0.67
NH Binomial (MCMC)	49.66	176.11	2.01	0.55
NH BB (MCMC)	52.96	213.76	2.44	0.66
NH Hurdle (MCMC)	53.99	223.51	2.56	0.69
Poisson w/Interactions (MLE)	34.51	108.07	1.21	0.33
NB w/Interactions (MLE)	43.01	184.22	2.11	0.57
Binomial w/Interactions (MLE)	34.53	105.44	1.21	0.33
H Poisson (MCMC)	34.57	106.11	1.21	0.33
H NB (MCMC)	46.73	185.24	2.12	0.57
H Binomial (MCMC)	**34.48**	**105.28**	**1.20**	**0.32**
H BB (MCMC)	47.50	186.05	2.13	0.57
H Hurdle (MCMC)	48.59	194.78	2.23	0.60

The choice of the model depends on the practitioner's expectations from the prediction. For modeling smaller counts, hierarchical negative binomial, beta-binomial, or hurdle models are preferable. For the models of the number of installations, the prediction accuracy of smaller counts is essential; zeros are especially vital since they automatically reduce the failure probability of a refrigerator to zero. Hence using the hurdle model might create an advantage in our case. Moreover, hierarchical models are more robust when data includes samples with few data points. Thus they will provide more stable predictions when new refrigerator models with a few observations are introduced.

Table 8.3: Comparison Statistics on Test Data
(NB=negative binomial, H=hierarchical, BB=beta-binomial, NH=non-hierarchical)

Model	MAE	RMSE	ARMSE	SRMSE
Poisson (MLE)	52.55	209.92	2.33	0.57
NB (MLE)	55.25	252.18	2.80	0.68
Binomial (MLE)	52.51	209.23	2.32	0.56
Hurdle (MLE)	55.51	253.34	2.82	0.68
NH Poisson (MCMC)	52.55	209.93	2.33	0.57
NH NB (MCMC)	55.29	252.08	2.80	0.68
NH Binomial (MCMC)	52.50	**209.22**	**2.32**	**0.56**
NH BB (MCMC)	54.61	250.39	2.78	0.68
NH Hurdle (MCMC)	56.26	260.73	2.89	0.71
Poisson w/Interactions (MLE)	$2 \cdot 10^6$	$2 \cdot 10^8$	$2 \cdot 10^6$	$6 \cdot 10^5$
NB w/Interactions (MLE)	$4 \cdot 10^{16}$	$4 \cdot 10^{18}$	$4 \cdot 10^{16}$	$1 \cdot 10^{16}$
Binomial w/Interactions (MLE)	56.59	225.29	2.50	0.61
H Poisson (MCMC)	56.89	234.79	2.61	0.63
H NB (MCMC)	**51.17**	219.87	2.44	0.60
H Binomial (MCMC)	55.67	221.08	2.45	0.60
H BB (MCMC)	52.03	222.34	2.47	0.60
H Hurdle (MCMC)	52.62	227.20	2.53	0.61

8.2 Comparisons of In-service Failure Models

This section discusses the accuracies of the models that are fitted for predicting the number of in-service refrigerator failures. The sampling of the non-hierarchical hurdle model was problematic and thus not included in the comparison. Additionally, there was no solution for interaction models; thus they were also discarded.

Table 8.4: Bayesian Comparison Statistics for Models of Number of Failures

Model	Loo	WAIC
Hierarchical Poisson	19950	19930
Hierarchical Negative Binomial	19953	19930
Hierarchical Binomial	**19935**	**19912**
Hierarchical Beta-Binomial	20072	20056
Hierarchical Hurdle	21643	21603
Non-Hierarchical Poisson	20750	20742
Non-Hierarchical Negative Binomial	20633	20616
Non-Hierarchical Binomial	20740	20728
Non-Hierarchical Beta-Binomial	20658	20673

The estimated loo and WAIC values of Bayesian models are presented in Table 8.4. The portion of problematic observations is less than 0.2% for all models. Thus models can be compared using estimated loo and WAIC. The predictive accuracies show that hierarchical models provide a better fit compared to their non-hierarchical versions. This might be expected since failure data is much more sparse than installation data. The percentage of zeros in the data is 65%. So, using the strength of hierarchical structure increases the prediction accuracy. The reader might notice that predictive accuracies of Poisson, negative binomial, and binomial models are nearly identical. This result supports the previous interpretation of diagnostic plots of these three models.

Table 8.5: Comparison Statistics on Train Data
(NB=negative binomial, H=hierarchical, BB=beta-binomial, NH=non-hierarchical)

Model	MAE	RMSE	ARMSE	SRMSE
Poisson (MLE)	0.34	0.89	1.91	0.76
NB (MLE)	0.35	0.92	1.96	0.78
Binomial (MLE)	0.34	0.89	1.91	0.76
NH Poisson (MCMC)	0.34	0.89	1.91	0.76
NH NB (MCMC)	0.35	0.92	1.96	0.78
NH Binomial (MCMC)	0.34	0.89	1.91	0.76
NH BB (MCMC)	0.36	0.94	2.01	0.80
H Poisson (MCMC)	0.29	0.66	1.41	0.56
H NB (MCMC)	0.29	0.68	1.45	0.58
H Binomial (MCMC)	**0.29**	0.66	1.41	0.56
H BB (MCMC)	0.31	0.71	1.52	0.61
H Hurdle (MCMC)	0.36	**0.59**	**1.26**	**0.51**

Both train and test accuracy results suggest that models that hierarchical models perform better than the models that utilize the MLE. This result indicates that Bayesian models can be preferable when the data are heterogeneous. The train accuracy results in Table 8.5 show that hierarchical models learn better than non-hierarchical ones. Also, test accuracies in Table 8.6 suggest that their predictive accuracy is better. The best MAE score is obtained by a hierarchical binomial model but squared error based metrics points to the hierarchical Poisson hurdle model. On the contrary, both estimated Bayesian compare metrics in Table 8.4 and test accuracies in Table 8.6 indicate that other hierarchical models predict better than the hurdle model. These results also designate the hurdle model is slightly overfitting.

Table 8.6: Comparison Statistics on Test Data
(NB=negative binomial, H=hierarchical, BB=beta-binomial, NH=non-hierarchical)

Model	MAE	RMSE	ARMSE	SRMSE
Poisson (MLE)	0.34	0.99	2.07	0.73
NB (MLE)	0.35	1.01	2.11	0.75
Binomial (MLE)	0.34	0.99	2.07	0.73
NH Poisson (MCMC)	0.34	0.99	2.07	0.73
NH NB (MCMC)	0.35	1.01	2.12	0.75
NH Binomial (MCMC)	0.34	0.99	2.07	0.73
NH BB (MCMC)	0.36	1.03	2.16	0.77
H Poisson (MCMC)	0.33	0.83	1.75	0.62
H NB (MCMC)	0.33	0.84	1.76	0.62
H Binomial (MCMC)	**0.33**	**0.83**	**1.75**	**0.62**
H BB (MCMC)	0.34	0.86	1.81	0.64
H Hurdle (MCMC)	0.40	1.20	2.32	0.82

The test accuracies indicate that all the hierarchical models except the hurdle model are almost identical. Although choosing any of these models does not make a difference in the predictive accuracy, we prefer to use the binomial model to predict the number of defective refrigerators since obtaining hazard probabilities is more interpretable in defective analysis. Furthermore, a more complex and suitable model can be obtained by decomposing hazard probabilities according to age covariate.

Chapter 9

Conclusion

This study provides a comparison of Bayesian models that are fitted to informative and less informative data and shows how the suitable structure depends on them. In the first part of the problem, we built both hierarchical and non-hierarchical Bayesian models for predicting the installation numbers. The data set for this part of the problem is informative enough, and the hierarchical structure does not make much difference. Comparing these models is challenging since the most stable Bayesian comparison methods are unreliable for the models that yield the best test accuracy. However, we prioritize increasing the predictive accuracy of zero counts since they are more impactful for the second part of the problem. Hence we recommend using the negative binomial hurdle model to predict the installation numbers. Furthermore, the second part of the problem emphasizes the value of the hierarchical structure.

Using the same structure for both parts portrays the consequences of the information amount in the data. The second dataset is considerably more sparse compared to the first one. This difference reflects in the models' performances. Besides the increased performance of the hierarchical models, we observed that their equivalence in the frequentist approach could not even find a solution. This result makes Bayesian models more preferable for practitioners. All the hierarchical models in the second part perform similarly except the hurdle model.

The hurdle model appeared to be unnecessarily complex for the second dataset. Although the similarity of the proposed model, we choose to use a hierarchical binomial model for predicting the defective numbers. We believe that using hazard probabilities is more interpretable for the problem, and their representation can be improved in future work.

In the training phase of the failure models, we used a training dataset which is the subset of the failure data. So, we used actual installation numbers to estimate the failure rates and trained our models accordingly. Here we may be concerned that our test accuracies of failure models are too optimistic since we do not consider the uncertainty that comes from the prediction of installation numbers. To obtain more realistic results, we can generate installation numbers for failure data and use these values in the training and testing phases. Furthermore, this procedure can be repeated for all binary combinations of installation and failure models to obtain a comparison of two-phased model structures. Of course, this comparison is quite computationally costly, and the total run time can be extremely long.

In future work, the prediction accuracy of models can be improved by implementing hybrid models. As a popular choice in Bayesian modeling, stacking or the Bayesian model averaging can be used when non of the competing models are fully representative. The study by Yao [31] indicates that stacking Bayesian models provides more robust predictions, and they can be implemented by adjusting the averaging weights with estimated loo or WAIC. Poor predictive results may indicate that there is unobserved heterogeneity in our data.

Some part of this variation can be expressed by properly adding the age covariate. The non-linearity inherent in age covariate can be modeled by an artificial neural network(ANN). Furthermore, a Bayesian ANN can improve the predictive accuracy more dramatically [32]. For modeling the hazard probabilities in the second part, a more complex structure can be used. We believe that the age of breakdown might affect the hazard probabilities; thus, representing those probabilities according to time can be more appropriate. On the other hand, data may

not be informative enough for such complex modeling. For implementing this survival model, data may need to be balanced using oversampling or undersampling methods.

Bibliography

[1] M. D. Lee, "How cognitive modeling can benefit from hierarchical bayesian models," *Journal of Mathematical Psychology*, vol. 55, no. 1, p. 1–7, 2011.

[2] B. E. Neuenschwander and M. Zwahlen, "Problems due to small samples and sparse data in conditional logistic regression analysis," *American Journal of Epidemiology*, vol. 152, no. 7, p. 688–689, 2000.

[3] A. Kotsialos, M. Papageorgiou, and A. Poulimenos, "Long-term sales forecasting using Holt-Winters and neural network methods," *Journal of Forecasting*, vol. 24, no. 5, p. 353–368, 2005.

[4] H. F. Carman, "Improving sales forecasts for appliances," *Journal of Marketing Research*, vol. 9, no. 2, p. 214, 1972.

[5] S. Kolassa, "Evaluating predictive count data distributions in retail sales forecasting," *International Journal of Forecasting*, vol. 32, no. 3, p. 788–803, 2016.

[6] L. R. Berry and M. West, "Bayesian forecasting of many count-valued time series," *Journal of Business & Economic Statistics*, vol. 38, no. 4, p. 872–887, 2019.

[7] D. Lambert, "Zero-inflated poisson regression, with an application to defects in manufacturing," *Technometrics*, vol. 34, no. 1, p. 1, 1992.

[8] D. A. Reda, I. H. Challoob, and S. H. Omran, "The use of time series to predict defective percentages is an applied study in the ishtar fireplace laboratory," *2020 2nd Al-Noor International Conference for Science and Technology (NICST)*, 2020.

[9] Y. Wang, Y. Ni, and X. Wang, "Real-time defect detection of high-speed train wheels by using bayesian forecasting and dynamic model," *Mechanical Systems and Signal Processing*, vol. 139, p. 106654, 2020.

[10] Y. Gao, W. Duan, and H. Rui, "Does social media accelerate product recalls? evidence from the pharmaceutical industry," *Information Systems Research*, p. 1–24, 2021.

[11] J. Reefhuis, O. Devine, J. M. Friedman, C. Louik, and M. A. Honein, "Specific SSRIs and birth defects: Bayesian analysis to interpret new data in the context of previous reports," *BMJ*, vol. 351, 2015.

[12] P. K. Dunn and G. K. Smyth, *Generalized Linear Models with Examples in R*. Springer, 2018.

[13] E. Cepeda-Cuervo and M. V. Cifuentes-Amado, "Double generalized beta-binomial and negative binomial regression models," *Revista Colombiana de Estadística*, vol. 40, no. 1, p. 141–163, 2017.

[14] A. C. Cameron and P. K. Trivedi, *Regression analysis of Count Data*. Cambridge University Press, 2013.

[15] R. Winkelmann, *Econometric Analysis of Count Data*. Springer, 2010.

[16] J. Mullahy, "Specification and testing of some modified count data models," *Journal of Econometrics*, vol. 33, no. 3, p. 341–365, 1986.

[17] C. X. Feng, "A comparison of zero-inflated and hurdle models for modeling zero-inflated count data," *Journal of Statistical Distributions and Applications*, vol. 8, no. 1, 2021.

[18] A. Gelman, D. Simpson, and M. Betancourt, "The prior can often only be understood in the context of the likelihood," *Entropy*, vol. 19, no. 10, p. 555, 2017.

[19] A. Gelman, "Bayes, jeffreys, prior distributions and the philosophy of statistics," *Statistical Science*, vol. 24, no. 2, 2009.

[20] P. Congdon, *Bayesian Hierarchical Models: With Applications Using R, second edition*. Routledge, 2021.

[21] P. Congdon, *Applied Bayesian hierarchical methods*. CRC Press, 2010.

[22] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, *Handbook of Markov Chain Monte Carlo* . CRC Press, 2011.

[23] A. Gelman, D. B. Rubin, J. B. Carlin, and H. S. Stern, *Bayesian Data Analysis*. Chapman & Hall, 2013.

[24] M. Betancourt, "Hierarchical modeling." `https://betanalpha.github.io/assets/case_studies/hierarchical_modeling.html`, 2020.

[25] M. Betancourt, "Identity crisis." `https://betanalpha.github.io/assets/case_studies/identifiability.html`, 2020.

[26] J. Gabry and M. Modrák, "Visual MCMC diagnostics using the bayesplot package." `https://mc-stan.org/bayesplot/articles/visual-mcmc-diagnostics.html`, 2022.

[27] A. Vehtari, A. Gelman, and J. Gabry, "Practical bayesian model evaluation using leave-one-out cross-validation and waic," *Statistics and Computing*, vol. 27, no. 5, p. 1413–1432, 2016.

[28] P.-C. Bürkner, "brms: An R package for Bayesian multilevel models using stan," *Journal of Statistical Software*, vol. 80, no. 1, 2017.

[29] A. Gelman, "Prior distributions for variance parameters in hierarchical models (comment on article by browne and draper)," *Bayesian Analysis*, vol. 1, no. 3, 2006.

[30] S. D. Team, "Stan functions reference." `https://mc-stan.org/docs/2_18/functions-reference/lkj-correlation.html`, 2022.

[31] Y. Yao, A. Vehtari, D. Simpson, and A. Gelman, "Using stacking to average Bayesian predictive distributions (with discussion)," *Bayesian Analysis*, vol. 13, no. 3, 2018.

[32] F. Liang, Q. Li, and L. Zhou, "Bayesian neural networks for selection of drug sensitive genes," *Journal of the American Statistical Association*, vol. 113, no. 523, p. 955–972, 2018.