3

Bayesian Linear Regression

Linear regression is one of the most common statistical approaches for modeling the relationship between a scalar dependent variable (or response) and one or more explanatory variables (or independent variables). It is the study of linear, additive relationships between variables. The methodology was the first type of regression analysis to be studied rigorously, and has been a topic of innumerable textbooks (Chatterjee and Hadi, 2015). Although it may seem to be too simple compared to some of the more modern statistical regression techniques described in later chapters of this book, linear regression is still considered as one of the most useful and powerful tools in practical applications. This chapter can serve as a good starting point for newer and more complex modeling approaches that we will discuss in the later chapters. Having a deep understanding of standard linear regression is of importance, since many fancy regression techniques can be viewed as generalizations or extensions of it.

3.1 Introduction

Let

$$\{x_{i1},...,x_{ip},y_i\}, \qquad i = 1,...,n,$$

represent n observation units, each of which consists of a measurement of the p-vector of predictors $(x_1,...,x_p)$ and a measurement of the response variable y. The multiple linear regression model takes the form

$$y_i = \beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip} + \varepsilon_i. \tag{3.1}$$

In this linear model (3.1), the relationship between y and $(x_1,...,x_p)$ is modeled using the linear predictor function $\mu = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p$, and a disturbance term or error variable ε. The unknown model parameters $(\beta_0,\beta_1,...,\beta_p)$ are estimated from the data. The model becomes a simple linear regression when $p = 1$. Linearity here is with respect to the unknown parameters. Sometimes the predictor function contains a nonlinear function of a predictor. For example, a polynomial regression of degree 3 is expressed as $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \varepsilon_i$. This model remains linear since it is linear in the parameter vector.

The linear model (3.1) can be written in a matrix form,

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon, \tag{3.2}$$

where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}, \quad \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}.$$

We begin our discussion by assuming that the errors are independent and normally distributed with mean zero and constant variance, i.e., the error term ε in (3.2) is assumed to be distributed as $N(0, \sigma^2\mathbf{I})$ with an unknown variance parameter σ^2.

In frequentist statistics, the parameters can be estimated using the maximum likelihood estimation (MLE) or the least squares method. Specifically, the likelihood function for the model (3.2) is,

$$L(\beta, \sigma^2|\mathbf{X}, \mathbf{y}) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \exp\left[-\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta)\right], \tag{3.3}$$

which yields the score equations

$$S_1(\beta, \sigma^2) = \frac{\partial \log L}{\partial \beta} = -\frac{1}{2\sigma^2}\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta), \tag{3.4}$$

$$S_2(\beta, \sigma^2) = \frac{\partial \log L}{\partial \sigma} = -\frac{n}{\sigma} + \frac{1}{2\sigma^3}(\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta). \tag{3.5}$$

Assuming $\mathbf{X}^T\mathbf{X}$ is of full rank and setting (3.4) and (3.5) to zero, we obtain the maximum likelihood estimators

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}, \tag{3.6}$$

and

$$\hat{\sigma}^2 = \frac{1}{n}(\mathbf{y} - \mathbf{X}\hat{\beta})^T(\mathbf{y} - \mathbf{X}\hat{\beta}). \tag{3.7}$$

Note that $\hat{\beta}$ is an unbiased estimator of β, and the least squares estimator of β is also $\hat{\beta}$ (Chatterjee and Hadi, 2015). However, $\hat{\sigma}^2$ is not an unbiased estimator of σ^2. The more commonly used estimator of σ^2, which is unbiased, is

$$S^2 = \frac{1}{n-p-1}(\mathbf{y} - \mathbf{X}\hat{\beta})^T(\mathbf{y} - \mathbf{X}\hat{\beta}).$$

3.2 Bayesian Inference for Linear Regression

We now consider Bayesian inference for the model (3.2). In Bayesian analysis, the inverse of the variance parameter plays an important role and is called the *precision*,

$\tau = \sigma^{-2}$. We shall use the precision τ in manipulating the distributions. Based on the assumption of the model, we have

$$\mathbf{y}|\boldsymbol{\beta},\tau \sim N(\mathbf{X}\boldsymbol{\beta},\tau^{-1}\mathbf{I}).$$

We further assume $\boldsymbol{\beta}$ and τ are independent. Therefore, the joint posterior distribution of the unknown parameters, thus, is

$$\pi(\boldsymbol{\beta},\tau|\mathbf{X},\mathbf{y}) \propto L(\boldsymbol{\beta},\tau|\mathbf{X},\mathbf{y})p(\boldsymbol{\beta})p(\tau),$$

where $p(\boldsymbol{\beta})$ and $p(\tau)$ are the priors for the parameters $\boldsymbol{\beta}$ and τ. Closed form of the posterior distributions of $\boldsymbol{\beta}$ and τ are only available under certain restricted prior distributions.

An important problem in Bayesian analysis is how to define the prior distribution. If prior information about the parameters is available, it should be incorporated in the prior distribution. If we have no prior information, we want that a prior distribution can be guaranteed to have a minimal influence on the inference. Noninformative prior distribution, for example $p(\boldsymbol{\beta}) \propto 1$, has always been appealing, since many real applications lack information on the parameters. However, the major drawback of noninformative prior is that it is not invariant for transformation of the parameters. See more discussions in Appendix B. We also refer to Gelman et al. (2014) for a comprehensive discussion on prior selection.

In INLA, we assume that the model is a latent Gaussian model, that is, we have to assign $\boldsymbol{\beta}$ a Gaussian prior. For the hyperparameter τ, we often assume a diffuse prior, a probability distribution with an extremely large variance. A typical prior choice for $\boldsymbol{\beta}$ and τ is

$$\boldsymbol{\beta} \sim N_{p+1}(\mathbf{c}_0,\mathbf{V}_0), \quad \tau \sim Gamma(a_0,b_0).$$

Here the prior of $\boldsymbol{\beta}$ is $p+1$-dimensional multivariate normal with known $\mathbf{c}_0$ and $\mathbf{V}_0$. We often assume that $\mathbf{V}_0$ is diagonal, which is equivalent to specifying separate univariate normal priors on the regression coefficients. The precision τ follows a dispersed gamma distribution with a known shape parameter a_0 and a known rate parameter b_0 (that is, we have mean a_0/b_0 and variance a_0/b_0^2). In linear regression, the gamma prior is conditionally conjugate for τ since the conditional posterior distribution, $p(\tau|\mathbf{X},\mathbf{y})$, is also in that class.

Although the posterior is intractable under these priors, it is straightforward to construct a blocked Gibbs sampling algorithm and be suitable for MCMC implementation (Gelman et al., 2014). Specifically, the algorithm iterates between the pair of conditional distributions:

$$\begin{cases} \pi(\boldsymbol{\beta}|\mathbf{X},\mathbf{y},\tau) \propto L(\boldsymbol{\beta},\tau|\mathbf{X},\mathbf{y})p(\boldsymbol{\beta}), \\ \pi(\tau|\mathbf{X},\mathbf{y},\boldsymbol{\beta}) \propto L(\boldsymbol{\beta},\tau|\mathbf{X},\mathbf{y})p(\tau). \end{cases}$$

Instead of MCMC simulations, the INLA approach provides approximations to the posterior marginals of the parameters which are both very accurate and extremely fast to compute (Rue et al., 2009). The marginal posterior $\pi(\tau|\mathbf{X},\mathbf{y})$ is approximated using

$$\tilde{\pi}(\tau|\mathbf{X},\mathbf{y}) \propto \left.\frac{\pi(\boldsymbol{\beta},\tau,\mathbf{X},\mathbf{y})}{\tilde{\pi}(\boldsymbol{\beta}|\tau,\mathbf{X},\mathbf{y})}\right|_{\boldsymbol{\beta}=\boldsymbol{\beta}^*(\tau)}, \tag{3.8}$$

which is the Gaussian approximation to the full conditional distribution of β evaluated in the mode $\beta^*(\tau)$ for a given τ. Expression (3.8) is equivalent to the Laplace approximation of a marginal posterior distribution (Tierney and Kadane, 1986), and it is exact when $\pi(\beta|\tau,\mathbf{X},\mathbf{y})$ is Gaussian.

Posterior marginals for the model parameters, $\tilde{\pi}(\beta_j|\tau,\mathbf{X},\mathbf{y})$, $j = 0,1,...,p$, are then approximated via numerical integration as:

$$\begin{aligned}\tilde{\pi}(\beta_j|\mathbf{X},\mathbf{y}) &= \int \tilde{\pi}(\beta_j|\tau,\mathbf{X},\mathbf{y})\tilde{\pi}(\tau|\mathbf{X},\mathbf{y})d\tau \\ &\approx \sum_k \tilde{\pi}(\beta_j|\tau_k,\mathbf{X},\mathbf{y})\tilde{\pi}(\tau_k|\mathbf{X},\mathbf{y})\Delta_k,\end{aligned}$$

where the sum is over values of τ with area weights Δ_k. For more technical details, we refer back to Chapter 2.

The approximate posterior marginals obtained from the INLA procedure can then be used to compute summary statistics of interest, such as posterior means, variances and quantiles. As a by-product of the main computations, INLA also computes other quantities of interest like *deviance information criterion* (DIC), marginal likelihoods, etc., which are useful to compare and validate models.

Let us look at an example of multiple linear regression of analyzing the air pollution data. The dataset has been discussed in Everitt (2006). The data were collected to investigate the determinants of pollution for 41 cities in the United States. Table 3.1 displays the variables being recorded and their descriptions. In this study, `SO2` level is considered as the dependent variable and the other six variables are considered as potential explanatory variables. Among these potential predictors, two of them are related to human ecology (`pop`, `manuf`) and four others are related to climate (`negtemp`, `wind`, `precip`, `days`). Note that the variable, `negtemp`, represents the negative value of average annual temperature. Using the negative values here is because all variables are such that high values represent a less attractive environment.

TABLE 3.1

Description of variables in the air pollution data.

Variable Name	Description	Codes/Values
SO2	sulfur dioxide content of air	micrograms per cubic meter
negtemp	negative value of average annual temperature	fahrenheit
manuf	number of manufacturing enterprises employing 20 or more workers	integers
pop	population size in thousands (1970 census)	numbers
wind	average annual wind speed	miles per hour
precip	average annual precipitation	inches
days	average number of days with precipitation per year	integers

Prior to performing a regression analysis on the data, it is useful to graph the da

in a certain way so that we can have an insight into the overall structure of them. Figure 3.1 presents a matrix of scatterplots for all variables on the upper triangular, and the correlation coefficients displayed on the lower triangle, with nonparametric kernel density plots for each variable on the main diagonal. The figure is created using the following R code:

```
library(ggplot2, GGally)
data(usair, package = "brinla")
pairs.chart <- ggpairs(usair[,-1], lower = list(continuous = "cor"),
    ↪ upper = list(continuous = "points", combo = "dot")) + ggplot2::
    ↪ theme(axis.text = element_text(size = 6))
pairs.chart
```

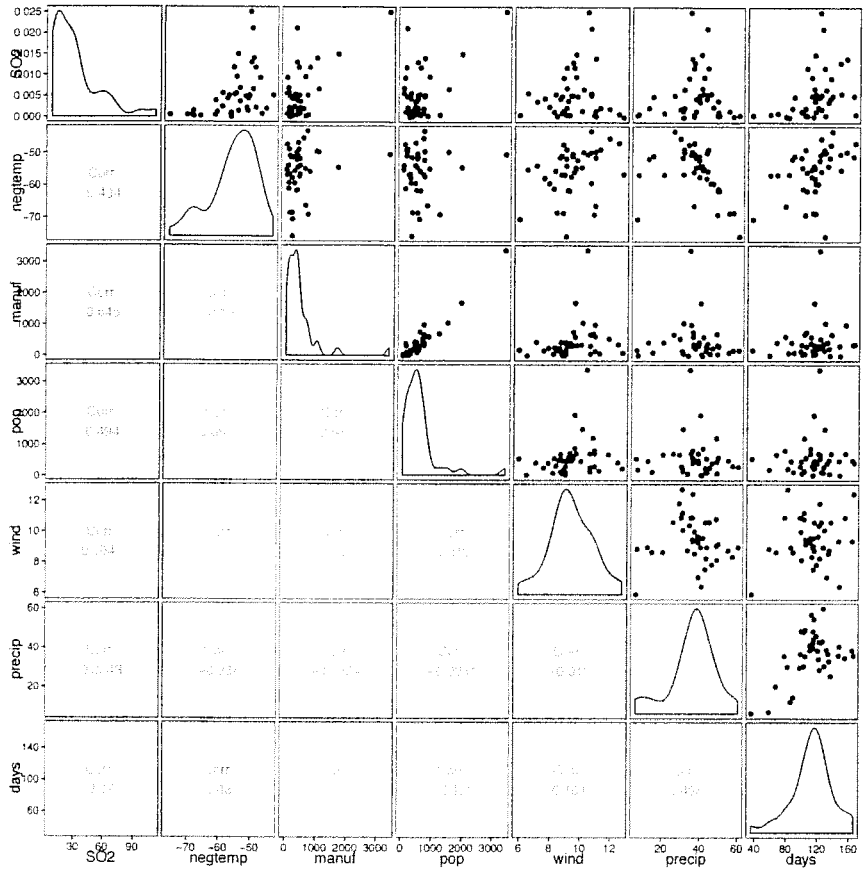

FIGURE 3.1
Scatterplot matrix of the variables in the air pollution data: The upper triangular displays the paired scatterplots; the lower triangle shows the paired correlation coefficients; and the main diagonal presents nonparametric kernel density plots for each variable.

From Figure 3.1 we notice that manuf and pop are highly correlated. Checking their sample correlation, we find that it is as high as 0.955. This phenomenon is known as multicollinearity, in which two or more predictors in a multiple regression model are highly correlated. In this situation the coefficient estimates of the multiple regression are very sensitive to slight changes in the data and to the addition or deletion of variables in the equation. The estimated regression coefficients often have large sampling errors, which affect both inference and prediction that is based on the model (Chatterjee and Hadi, 2015).

To avoid the multicollinearity issue, we use a simple approach by keeping only one of the two variables, manuf, in our regression analysis. A more sophisticated method to deal with multicollinearity is addressed in Section 3.7.

For comparison purposes, we begin the analysis with the conventional maximum likelihood method. We first fit a regression model with the five predictors, negtemp, mauf, wind, precip, and days:

```
usair.formula1 <-  SO2 ~ negtemp + manuf + wind + precip + days
usair.lm1 <- lm(usair.formula1, data = usair)
round(coef(summary(usair.lm1)), 4)
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 135.7714    50.0610  2.7121   0.0103
negtemp       1.7714     0.6366  2.7824   0.0086
manuf         0.0256     0.0046  5.5544   0.0000
wind         -3.7379     1.9444 -1.9224   0.0627
precip        0.6259     0.3885  1.6111   0.1161
days         -0.0571     0.1748 -0.3265   0.7460
```

The estimated residual standard error, $\hat{\sigma}$, is given by:

```
round(summary(usair.lm1)$sigma, 4)
```

```
[1] 15.79
```

The results for this model show that the predictors negtemp and manuf are significant predictors while the predictors wind, precip, and days are not. We now fit Bayesian models using INLA with the default priors. In the INLA package, the default choice of priors for $\beta_j, j = 0, \ldots, p$

$$\beta_j \sim N(0, 10^6), \quad j = 0, \ldots, p,$$

and the prior for the variance parameter is defined internally in terms of logged precision, $\log(\tau)$. It follows a log gamma distribution,

$$\log(\tau) \sim \text{logGamma}(1, 10^{-5}).$$

We fit a Bayesian linear regression using INLA with the following code:

```
library(INLA)
usair.inla1 <- inla(usair.formula1, data = usair, control.compute =
    ↪ list(dic = TRUE, cpo = TRUE))
```

The inla function returns an object, here named usair.inla1, which has a clas attribute, inla. This is a list containing a lot of objects which can be explore with names(usair.inla1). For example, the summary of the fixed effects can b obtained by the command:

```
round(usair.inla1$summary.fixed, 4)
```

```
              mean      sd 0.025quant 0.5quant 0.975quant     mode kld
(Intercept) 135.4892 50.0629    36.6158 135.4955   234.1778 135.5116   0
negtemp       1.7690  0.6370     0.5111   1.7690     3.0247   1.7692   0
manuf         0.0256  0.0046     0.0165   0.0256     0.0347   0.0256   0
wind         -3.7229  1.9424    -7.5570  -3.7234     0.1080  -3.7241   0
precip        0.6249  0.3888    -0.1429   0.6249     1.3913   0.6249   0
days         -0.0567  0.1749    -0.4020  -0.0567     0.2882  -0.0567   0
```

The summary of the hyperparameter is obtained by

```
round(usair.inla1$summary.hyperpar, 4)
```

```
                                mean    sd 0.025quant 0.5quant 0.975quant  mode
Precision
for the Gaussian observations 0.0042 9e-04     0.0026   0.0042     0.0063 0.004
```

Summaries of these posterior distributions include posterior means and 95% credible intervals, which can be used as Bayesian alternatives to the maximum likelihood estimates and 95% confidence intervals, respectively. For example, the posterior mean of the coefficient for `negtemp` is 1.7690, and the 95% credible interval is (0.5111, 3.0247). These indicate that, with very high probability, `negtemp` is positively associated with the response, `SO2`. Unlike confidence intervals, which are calculated by assuming large sample approximations, Bayesian interval estimates are typically appropriate in small samples. More importantly, the Bayesian 95% credible interval estimates have an intuitively appealing interpretation as the interval containing the true parameter with 95% probability. This interpretation is often preferable to that of the 95% confidence interval, which is the range of values containing the true parameter 95% of the time in repeated sampling.

Another simple way to look at the result from an `inla` object is to use `summary` function, which produces default summaries of the results of the `inla` fitting function:

```
summary(usair.inla1)
```

The result summaries, which we do not display here, output the summary statistics of posteriors distributions of the fixed effects and the hyperparameters for the model including posterior means, standard deviations, the quartiles and others. Some model statistics, such as marginal log-likelihood, and the model fitting index, DIC (when we specify `DIC = TRUE` in `inla`), are printed. Users can also selectively print certain model results by their needs by making use of the `$` sign, as we have shown above. Some other useful information, for example, posterior marginal distributions of the fixed effects parameters and the hyperparameters can be obtained by the commands, `usair.inla1$marginals.fixed` and `usair.inla1$marginals.hyperpar`.

The `INLA` library includes a set of functions to operate on marginal distributions. The commonly used functions include `inla.dmarginal`, `inla.pmarginal`, `inla.qmarginal`, `inla.mmarginal`, and `inla.emarginal` to compute the density, distribution, quantile function, mode, and expected values of marginals, respectively. The function `inla.rmarginal` is used to generate random numbers, and the function `inla.tmarginal` can be used to transform a given marginal distribution. Here we show an example of how to make use of the functions.

By default, the posterior summaries of the precision τ is outputted from an `inla` object. However, we are often interested in the posterior mean of σ. The estimate can be obtained by using the `inla.emarginal` function:

```
inla.emarginal(fun = function(x) 1/sqrt(x), marg = usair.inla1$
    ↪ marginals.hyperpar$`Precision for the Gaussian observations`)
```

```
[1] 15.66331
```

Comparing the estimated residual standard error from the conventional maximum likelihood method, we obtain very similar results. For the user's convenience, we have written an R function `bri.hyperpar.summary` for producing the summary statistics of hyperparameters in terms of σ in our `brinla` package:

```
library(brinla)
round(bri.hyperpar.summary(usair.inla1),4)
```

```
                                    mean     sd  q0.025    q0.5 q0.975    mode
SD for the Gaussian observations 15.6617 1.8379 12.5321 15.4871 19.759 15.1513
```

We want to further look at the plot of the posterior distribution of σ. The function `bri.hyperpar.plot` in `brinla` can be applied to produce this posterior density directly:

```
bri.hyperpar.plot(usair.inla1)
```

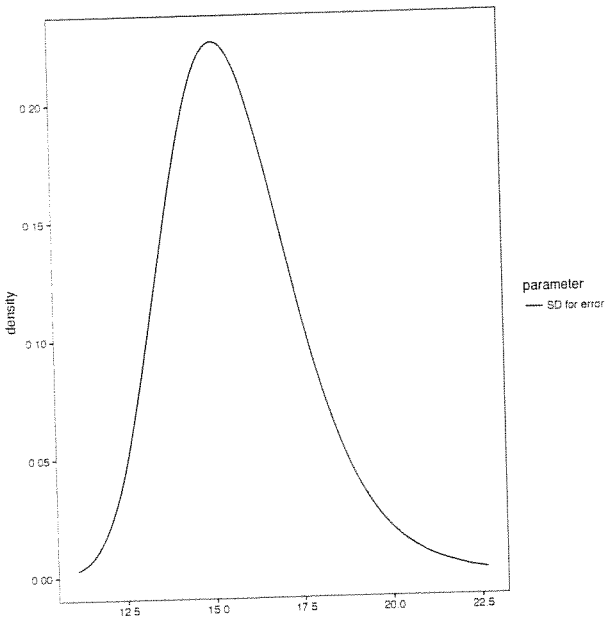

FIGURE 3.2
Posterior density for the parameter σ in the US air pollution study.

In Figure 3.2, we see a slightly right-skewed posterior distribution for σ.

The INLA program allows the user to change the prior for the regression parameters. Suppose that we have certain prior information for the intercept β_0 and the coefficients of `negtemp` and `wind`. For example, we assume that $\beta_0 \sim N(100, 100)$, $\beta_{negtemp} \sim N(2, 1)$, and $\beta_{wind} \sim N(-3, 1)$. The prior specification can be achieved using the option `control.fixed` in `inla` function. By default, a diffuse gamma prior is assumed on the precision parameter τ. If we want to specify, for instance, a log-normal prior to τ (equivalent to assuming a normal prior on the logarithm of τ), this can be specified using the option `control.family`:

```
usair.inla2 <- inla(usair.formula1, data = usair, control.compute =
    ↪ list(dic = TRUE, cpo =TRUE), control.fixed = list(mean.
    ↪ intercept = 100, prec.intercept = 10^(-2), mean = list(negtemp
    ↪ = 2, wind = -3, default =0), prec = 1), control.family = list(
    ↪ hyper = list(prec = list(prior="gaussian", param =c(0,1)))))
```

Note that here we change the priors for the intercept as well as two fixed parameters for `negtemp` and `wind`. The statement `mean = list(negtemp = 2, wind = -3, default = 0)` assigns prior mean equal to 2 for `negtemp`, -3 for `wind`, and zero means for all other parameters of the remaining predictors, using `list`. A `list` has to be also specified for `prec` if we have different precision assumptions. Certainly, users need to be careful in changing priors. The model selection and checking methods discussed in Section 3.4 can be used for comparing models with different priors.

3.3 Prediction

Suppose we apply the regression model to a new set of data, for which we have observed the vector of explanatory variables $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_p)^T$, and wish to predict the outcome, $\tilde{y}$. Typically, the Bayesian prediction is made based on the *posterior predictive distribution*, $p(\tilde{y}|\mathbf{y})$. Here "posterior" means that it is conditional on the observed $\mathbf{y}$, and "predictive" means that it is a prediction for $\tilde{y}$. Let $\theta = (\beta, \tau)$. We have

$$\begin{aligned} p(\tilde{y}|\mathbf{y}) &= \frac{p(\tilde{y}, \mathbf{y})}{p(\mathbf{y})} = p(\mathbf{y})^{-1} \int p(\tilde{y}|\theta) p(\mathbf{y}|\theta) p(\theta) \mathrm{d}\theta \\ &= p(\mathbf{y})^{-1} \int p(\tilde{y}|\theta) p(\theta|\mathbf{y}) p(\mathbf{y}) \mathrm{d}\theta \\ &= \int p(\tilde{y}|\theta) p(\theta|\mathbf{y}) \mathrm{d}\theta. \end{aligned}$$

The analytic form of the posterior predictive distribution in most regression models is not available. In conventional Bayesian analysis, the prediction can be done by posterior predictive simulation, i.e., drawing random samples from $p(\tilde{y}|\mathbf{y})$.

Going back to the air pollution example, suppose that we have the following new observations:

```
new.data <- data.frame(negtemp = c(-50, -60, -40), manuf = c(150, 100,
```

```
    ↪ 400), pop = c(200, 100, 300), wind = c(6, 7, 8), precip = c
    ↪ (10, 30, 20), days = c(20, 100, 40))
```

To predict SO2 from the MLE fit, `usair.lm1`, in R, we run:

```
predict(usair.lm1, new.data, se.fit = TRUE)
```

```
$fit
       1        2        3
33.72743 18.94993 55.47696

$se.fit
        1         2         3
14.936928  5.329492 17.639438

$df
[1] 35

$residual.scale
[1] 15.78998
```

The R output includes a vector of predictions (`$fit`), a vector of standard error of predicted means (`$se.fit`), the degrees of freedom for residuals (`$df`), and the residual standard deviation (`$residual.scale`).

In the `INLA` library, there is no function "predict" as for `lm` in R. However, we do not need a posterior predictive simulation like in MCMC approaches. Predictions can be done as a part of the model fitting itself in INLA. As prediction is the same as fitting a model with some missing data, we need to set the response variables "`y[i] = NA`" for those "observations" we want to predict. The prediction in INLA is implemented through the following R code:

```
usair.combined <- rbind(usair, data.frame(SO2 = c(NA, NA, NA), new.
    ↪ data))
usair.link <- c(rep(NA, nrow(usair)), rep(1, nrow(new.data)))
usair.inla1.pred <- inla(usair.formula1, data = usair.combined,
    ↪ control.predictor = list(link = usair.link))
usair.inla1.pred$summary.fitted.values[(nrow(usair)+1):nrow(usair.
    ↪ combined),]
```

```
                        mean       sd 0.025quant 0.5quant 0.975quant     mode
fitted.predictor.42 33.65338 14.938107   4.191079 33.65547   63.09751 33.65951
fitted.predictor.43 18.92744  5.329609   8.416159 18.92807   29.43290 18.92930
fitted.predictor.44 55.40423 17.644354  20.605306 55.40629   90.18455 55.41030
```

Note that we set the `control.predictor` option as `control.predictor = list(link = usair.link)`, where the object `usair.link` is set to be a vector of `NA` if the corresponding response is observed in the original dataset and `1` if the corresponding response is missing (and is to be predicted). The summary statistics of the predicted responses from INLA show concordance with the results from the MI method.

3.4 Model Selection and Checking

In Chapter 1, we briefly discussed model selection and checking for a Bayesian model. Here we show the details on how to implement the analysis in INLA using the air pollution data example.

3.4.1 Model Selection by DIC

In regression analysis, we often want to find a reduced model with the best subset of the variables from the full model. The model selection in frequentist analysis is commonly based on Akaike information criterion (AIC), a MLE-based criterion. Back to the air pollution data example, a stepwise model selection procedure using AIC can be implemented by the function `stepAIC` in R library `MASS`:

```
library(MASS)
usair.step <- stepAIC(usair.lm1, trace = FALSE)
usair.step$anova
```

```
Stepwise Model Path
Analysis of Deviance Table

Initial Model:
SO2 ~ negtemp + manuf + wind + precip + days

Final Model:
SO2 ~ negtemp + manuf + wind + precip


    Step Df Deviance Resid. Df Resid. Dev      AIC
1                           35   8726.322 231.7816
2 - days  1 26.57448        36   8752.897 229.9063
```

It turns out that the variable, `days`, is dropped from the full model. The final reduced model includes the four predictors, `negtemp`, `manuf`, `wind`, and `precip`. Let us fit the final reduced model:

```
usair.formula2 <- SO2 ~ negtemp + manuf + wind + precip
usair.lm2 <- lm(usair.formula2, data = usair)
round(coef(summary(usair.lm2)), 4)
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 123.1183    31.2907  3.9347   0.0004
negtemp       1.6114     0.4014  4.0148   0.0003
manuf         0.0255     0.0045  5.6150   0.0000
wind         -3.6302     1.8923 -1.9184   0.0630
precip        0.5242     0.2294  2.2852   0.0283
```

In Bayesian analysis, DIC, a generalization of AIC, is one of the most popular measures for Bayesian model comparison, which is defined as the sum of a measure of goodness of fit plus a measure of model complexity (Spiegelhalter et al., 2002). In the `INLA` library, the `dic=TRUE` flag makes the `inla()` function compute the model's DIC. The model with the lower DIC provides the better trade off between fit and

model complexity. Unfortunately, there is no function available for stepwise model selection by DIC in the INLA library. In this example, there are only five predictors in the full model. We may perform a backward elimination procedure using DIC (i.e., manually eliminating variables based on DIC). In this air pollution study, it turns out that the final model with four predictors, negtemp, mauf, wind, and precip, has the lowest DIC (=348.57), which is concordant with the above model selection result using the frequentist approach:

```
usair.inla3 <- inla(usair.formula2, data = usair, control.compute =
    ↪ list(dic = TRUE, cpo = TRUE))
round(usair.inla3$summary.fixed, 4)
```

```
              mean      sd 0.025quant 0.5quant 0.975quant     mode kld
(Intercept) 122.9366 31.2837    61.1617 122.9406   184.5970 122.9507   0
negtemp       1.6102  0.4016     0.8172   1.6102     2.4019   1.6103   0
manuf         0.0255  0.0045     0.0165   0.0255     0.0344   0.0255   0
wind         -3.6168  1.8905    -7.3480  -3.6172     0.1111  -3.6179   0
precip        0.5239  0.2296     0.0706   0.5239     0.9766   0.5240   0
```

The cpo=TRUE flag will be discussed in a later section. Comparing the DICs between two Bayesian models, we have a smaller DIC for the last model:

```
c(usair.inla1$dic$dic, usair.inla3$dic$dic)
```

```
[1] 350.6494 348.5703
```

From the output, regression coefficients of the variables negtemp, mauf, and precip are significantly different from zero (in the Bayesian sense). That is, the 95% credible intervals of these coefficients do not contain zero. They have high posterior probabilities of being positively associated with the response SO2. The variable wind is negatively associated with SO2, but is not significant. Model coefficients can be interpreted as follows. For example, the posterior mean of precip, 0.5239, means that for every additional inch of average annual precipitation we expect the sulfur dioxide content of air to increase 0.5239 micrograms per cubic meter, when other covariates are fixed. The 95% credible interval for precip, is (0.0706, 0.9766), which contains the true parameter of precip with 95% probability.

3.4.2 Posterior Predictive Model Checking

Checking the model fit is critical in statistical analysis. In Bayesian analysis, model assessment is often based on posterior predictive checks or leave-one out cross-validation predictive checks. Held et al. (2010) compared two approaches for estimating Bayesian models using MCMC and INLA. Bayesian model posterior predictive check was originally proposed by Gelman et al. (1996). The key concept of such a check is the posterior predictive distribution of a replicate observation y_i^* which has density

$$p(y_i^*|\mathbf{y}) = \int p(y_i^*|\theta)p(\theta|\mathbf{y})d\theta.$$

The corresponding *posterior predictive p-value*,

$$p(y_i^* \leq y_i|\mathbf{y}),$$

is used as a measure of model fit (Meng, 1994). Extreme posterior predictive p-values ("extreme" means that p-value is very close to 0 or 1 here) can be used to identify observations that diverge from the assumed model.

In the `INLA` package, the posterior predictive p-value can be obtained by the R function `inla.pmarginal`, which returns the distribution function of marginals obtained by `inla`. The following R code generates the histogram of posterior predictive p-values for the reduced final model in the US air pollution study.

```
usair.inla3.pred <- inla(usair.formula2, data = usair, control.
    ↪ predictor = list(link = 1, compute = TRUE))
post.predicted.pval <- vector(mode = "numeric", length = nrow(usair))
for(i in (1:nrow(usair))) {
  post.predicted.pval[i] <- inla.pmarginal(q=usair$SO2[i], marginal =
      ↪ usair.inla3.pred$marginals.fitted.values[[i]])
}
hist(post.predicted.pval, main="", breaks = 10, xlab="Posterior
    ↪ predictive p-value")
```

Figure 3.3 shows that many posterior predictive p-values are close to 0 or 1. However, one drawback about interpreting posterior predictive p-values is that they could not have a uniform distribution even if the data come from the true model. See Hjort et al. (2006); Marshall and Spiegelhalter (2007) for the details. From the scatterplot matrix of the air pollution data (Figure 3.1), the response, `SO2`, is right-skewed and the predictor, `manuf`, has a very large variance and contains some outliers. The posterior predictive p-values could be affected by the nature of the data. So, although the plot of the posterior predictive p-values is not satisfied, we want to further check the model using other model assessment methods.

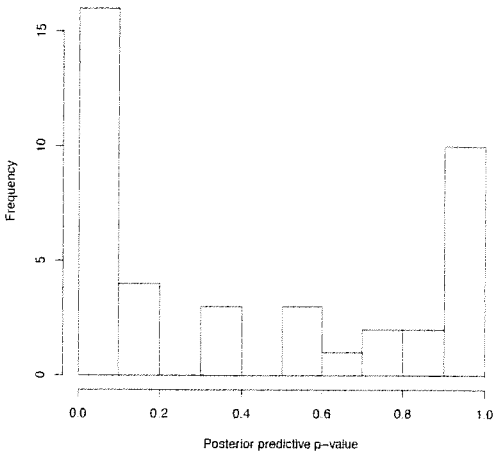

FIGURE 3.3
Histogram of the posterior predictive p-values for the reduced final model in the US air pollution study.

3.4.3 Cross-Validation Model Checking

The other methods based on the predictive distribution are the leave-one-out cros validation. Two quantities, *conditional predictive ordinate* (CPO) and *probability i tegral transform* (PIT), are used for evaluating the goodness of the model:

$$\text{CPO}_i = p(y_i|\mathbf{y}_{-i}),$$
$$\text{PIT}_i = p(y_i^* \leq y_i|\mathbf{y}_{-i}).$$

Here $\mathbf{y}_{-i}$ denotes the observations $\mathbf{y}$ with the i^{th} observation omitted. Note that only difference between PIT and the posterior predictive p-value is that PIT is co puted based on $\mathbf{y}_{-i}$ rather than $\mathbf{y}$.

In INLA, these quantities are computed without rerunning the model for e observation in turn (Held et al., 2010). To obtain CPOs and PITs, we need to s ply add the argument `control.compute = list(cpo = TRUE)` into `inla` fu tion. For example, in our resulting object `usair.inla3` for the final reduced mc using INLA, we can find the predictive CPOs and PITs using the comma `usair.inla3$cpo$cpo` and `usair.inla3$cpo$pit`. Held et al. (2010) showed merical problems may occur when CPOs and PITs are computed using INLA. Tl are internal checks in the INLA program for the potential problems, which app as `usair.inla3$cpo$failure`. It is a vector containing 0 or 1 for each observat A value equal to 1 indicates that the estimate of CPO or PIT is not reliable for corresponding observation. In our example, we can check if there are any failures

```
sum(usair.inla3$cpo$failure)
```

```
[1] 0
```

So, there is no issue of the computation of CPOs and PITs in the fit `usair.in` The uniformity of the PIT values indicates that the predictive distributions matcl observations from the data and thus it is an indication of a well-fitted model (Die et al., 1998; Gneiting et al., 2007). We now plot the histogram and the uniform plot of PITs.

```
hist(usair.inla3$cpo$pit, main="", breaks = 10, xlab = "PIT")
qqplot(qunif(ppoints(length(usair.inla3$cpo$pit))), usair.inla3$cp
    ↪ pit, main = "Q-Q plot for Unif(0,1)", xlab = "Theoretical
    ↪ Quantiles", ylab = "Sample Quantiles")
qqline(usair.inla3$cpo$pit, distribution = function(p) qunif(p), p
    ↪ = c(0.1, 0.9))
```

Figure 3.4 shows that the distribution of the PITs is close to a uniform distribu suggesting that the model reasonably fits the data. Note that the PIT histogr much closer to a uniform distribution than the corresponding posterior pred histogram shown in Figure 3.3.

If we think of the product of all of the CPO values as a "pseudo marginal l hood," this gives a cross-validatory summary measure of fit. The *log pseudo mai likelihood* (LPML), proposed by Geisser and Eddy (1979), is simply the log c

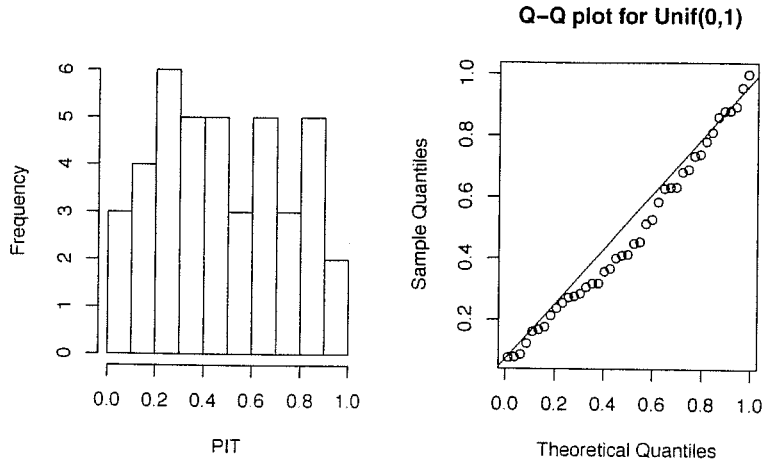

FIGURE 3.4
Histogram and uniform Q-Q plot of the cross-validated PIT for the reduced final model in the US air pollution study.

measure,

$$LPML = \log\left\{\prod_{i=1}^{n} p(y_i|\mathbf{y}_{-i})\right\} = \sum_{i=1}^{n} \log p(y_i|\mathbf{y}_{-i}) = \sum_{i=1}^{n} \log \text{CPO}_i,$$

which is often used as an alternative measure for DIC. Draper and Krnjajic (2007, Sec. 4.1) have shown that DIC approximates the LPML for approximately Gaussian posteriors. LPML remains computationally stable (Carlin and Louis, 2008). Unlike DIC, a model with a larger LPML is better supported by the data. Let us compute the LPMLs for the full model and reduced model in the air pollution study:

```
LPML1 <- sum(log(usair.inla1$cpo$cpo))
LPML3 <- sum(log(usair.inla3$cpo$cpo))
c(LPML1, LPML3)
```

```
[1] -176.9495 -175.6892
```

LPML for the reduced model is larger than that for the full model, indicating that the reduced model is preferred. Oftentimes, we can also perform a graphical analysis of a point-wise comparison of CPOs to choose a model.

```
plot(usair.inla1$cpo$cpo, usair.inla3$cpo$cpo, xlab="CPO for the full
    ↪ model", ylab="CPO for the reduced model")
abline(0,1)
```

Figure 3.5 shows a scatterplot of the pointwise comparison of CPOs between the full model and the reduced model, along with a reference line marking where the values are equal for the two models, in the US air pollution study. Since larger

4

Generalized Linear Models

Generalized linear models (GLMs), originally formulated by Nelder and Baker (2004), provide a unifying family of linear models that is widely used in practical regression analysis. The GLMs generalize ordinary linear regression by allowing the models to be related to the response variable via a link function and by allowing the magnitude of the variance of each measurement to be a function of its predicted value. Thus, these models allow for describing response variables that have an error distribution other than normal. They avoid having to select certain transformations of the data to achieve the possibly conflicting objects of normality, linearity and/or homogeneity of variance. Commonly used GLMs include logistic regression for binary data and Poisson regression or negative-binomial regression for count data.

4.1 GLMs

Let us start from a review of the exponential family of distributions. In statistics, the distribution of a random variable Y belongs to an exponential family if its probability density function (or probability mass function for the case of a discrete distribution) can be written in the form

$$f(y|\theta,\phi) = \exp\left\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right\}, \tag{4.1}$$

where $\theta = g(\mu)$, called the *canonical parameter*, is a function of the expectation $\mu \equiv E(Y)$ of Y, and the canonical link function $g(\cdot)$ does not depend on ϕ. The parameter $\phi > 0$, called the *dispersion parameter*, represents the scale of the distribution. The functions $a(\cdot)$, $b(\cdot)$, and $c(\cdot)$ are known functions that vary from one distribution to another.

The exponential families include many of the common distributions, including the normal, inverse Gaussian, exponential, gamma, Bernoulli, binomial, multinomial, Poisson, chi-squared, Wishart, Inverse Wishart and many others. Here we look at a few typical distributions in detail.

Let Y be normally distributed with mean μ and variance σ^2. Putting the normal distribution into the form of equation (4.1) requires some algebraic manipulation,

$$f(y|\theta,\phi) = \exp\left\{\frac{y\theta - \theta^2/2}{\phi} - \frac{1}{2}\left[\frac{y^2}{\phi} + \log(2\pi\phi)\right]\right\},$$

where $\theta = g(\mu) = \mu$, $\phi = \sigma^2$, $a(\phi) = \phi$, $b(\theta) = \theta^2/2$, and $c(y,\phi) = -(y^2/\phi + \log(2\pi\phi))/2$.

Now let us consider the binomial distribution, where Y is the number of "successes" in n independent binary trials, and μ is the probability of success in an individual trial. The probability mass function of Y is $f(y|\mu) = \binom{n}{y}\mu^y(1-\mu)^{n-y}$. Written as an exponential family, we have

$$f(y|\theta,\phi) = \exp\left\{y\theta - n\log(1+\exp\theta) + \log\binom{n}{y}\right\},$$

where $\theta = g(\mu) = \log\left(\frac{\mu}{1-\mu}\right)$, $\phi = 1$, $a(\phi) = 1$, $b(\theta) = n\log(1+\exp\theta)$, and $c(y,\phi) = \log\binom{n}{y}$.

The third example is Poisson distribution, which is used to model count data. It is appropriate for applications that involve counting the number of times a random event occurs in a given amount of time, distance, area, etc. Its probability mass function is $f(y|\mu) = \exp(-\mu)\mu^y/y!$, which can be rewritten as

$$f(y|\theta,\phi) = \exp(y\theta - \exp(\theta) - \log y!).$$

Here $\theta = \log(\mu)$, $\phi = 1$, $a(\phi) = 1$, $b(\theta) = \exp(\theta)$ and $c(y,\phi) = -\log y!$.

A key property of the exponential families is that the distributions have mean

$$E(Y) \equiv \mu = b'(\theta)$$

and variance

$$Var(Y) = a(\phi)b''(\theta).$$

Note that $b'(\cdot)$ is the inverse of the canonical link function. The mean of the distribution is a function of θ only, while the variance of the distribution is a product of functions of the location parameter θ and the scale parameter ϕ. In GLM, the $b''(\theta)$ is called the *variance function* to describe how the variance relates to the mean. Table 4.1 shows the link functions and their inverses, as well as the variance function for some common distributions.

TABLE 4.1
Link functions, their inverses, and variance functions for some common distributions.

Family	$\theta = g(\mu)$	$\mu = g^{-1}(\theta)$	Variance Function
Normal	μ	θ	1
Poisson	$\log\mu$	$\exp(\theta)$	μ
Binomial	$\log(\mu/(1-\mu))$	$\exp(\theta)/(1+\exp(\theta))$	$\mu(1-\mu)$
Gamma	μ^{-1}	θ^{-1}	μ^2
Inverse Gaussian	μ^{-2}	$\theta^{-1/2}$	μ^3

A GLM provides a unified modeling framework for many commonly used statistical models. Here we define the model in terms of a set of the observations $y_1,\ldots,$ which are regarded as realizations of random variables $Y_1,\ldots,Y_n$. There are the following three components in a GLM:

Random Component: The dependent variables, Y_i's, are assumed to be generated from a particular distribution in the exponential family (4.1).

Linear Predictor: That is a linear combination of the predictors

$$\theta_i = \beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip} = \mathbf{x}_i^T \boldsymbol{\beta},$$

where $\boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_p)^T$, $\mathbf{x}_i = (1, x_{1i}, ..., x_{ip})^T$, and x_{ij}, $j = 1, ..., p$ is the value of the j^{th} covariate for the i^{th} observation, as we have seen in a linear regression in Chapter 3.

Link Function: The expectation of the response variable, $\mu_i \equiv E(Y_i)$ and the linear predictor are related through a link function $g(\cdot)$:

$$g(\mu_i) = \theta_i.$$

Note that in most applications, the so-called *natural link function* is used, i.e., $g(\cdot) = b'(\cdot)$.

The GLM covers a large class of regression models, such as normal linear regression, logistic and probit regression, Poisson regression, negative-binomial regression and gamma regression. The classical estimation method for GLMs is maximum likelihood. There are several excellent textbooks discussing theory and applications for GLMs from a frequentist point of view; see for example, McCullagh and Nelder (1989); Lindsey (1997); Dobson and Barnett (2008). These books provide a rich collection of maximum likelihood estimation methods, hypothesis testing, real case studies.

For Bayesian analysis, MCMC is the common choice, which requires generating samples from posterior distributions. INLA treats a wide range of GLMs in a unified manner, thus allowing for greater automation of the inference process. In the rest of the chapter, we will discuss a few popular GLMs and demonstrate real case studies by applying the INLA method.

4.2 Binary Responses

In many applications, the response variable takes one of only two possible values representing success and failure, or more generally the presence or absence of an attribute of interest. Logistic regression, as a special case of GLMs, has a long tradition with widely varying applications to model such data. It is used to estimate the probability of a binary response based on one or more predictor variables.

Let Y be Bernoulli distributed with success probability $P(Y = 1) = \pi$. Its density is given by

$$f(y) = \exp\left\{ y \log\left(\frac{\pi}{1-\pi}\right) + \log(1-\pi) \right\}.$$

The distribution belongs to the exponential family, with canonical parameter θ equal to the logit of π, i.e., $\log(\pi/(1-\pi))$, dispersion parameter $\phi = 1$. Its mean is π and variance function is $\pi(1-\pi)$. The canonical link function, the logit link, leads to the classical logistic regression model:

$$\begin{cases} Y_i \sim \text{Bernoulli}(\pi_i), \\ \text{logit}(\pi_i) = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}. \end{cases}$$

Sometimes, the logit link function can be replaced by the probit link, which is the inverse of the standard normal distribution function, $\Phi^{-1}(\cdot)$. It has been shown that the logit and probit link functions behave similarly except the case for extreme probabilities (Agresti, 2012).

Here we analyze low birth weight data to illustrate the use of logistic regression. The dataset has been presented in Hosmer and Lemeshow (2004). The dataset contains information on 189 births to women seen in the obstetric clinic, where data were collected as part of a larger study at Baystate Medical Center in Springfield, Massachusetts. The response variable `LOW` is a binary outcome indicating birth weight less than 2500 grams, which has been of concern to physicians for years. A woman's behavior during pregnancy, such as smoking habits, receiving prenatal care can greatly change the chances of carrying the baby to term, and thus, of delivering a baby of normal birth weight. The variables that are potentially associated with low birth weight are recorded in the study, given in Table 4.2. The goal of this study was to determine whether some or all of these variables were risk factors in the clinic population being treated by the medical center.

TABLE 4.2

Code sheet for the variables in the low birth weight data.

Variable Name	Description	Codes/Values
LOW	indicator of low birth weight	0 = $\geq$ 2500g 1 = $<$ 2500g
AGE	age of mother	years
LWT	weight of mother at last menstrual period	pounds
RACE	race of mother	1 = white 2 = black 3 = other
SMOKE	smoking status during pregnancy	0=no 1 = yes
HT	history of hypertension	0 = no 1 = yes
UI	presence of uterine irritability	0 =no 1 = yes
FTV	number of physician visits during the first trimester	counts

Data were collected on 189 women, 59 of whom had low birth weight babies and 130 of whom had normal birth weight babies. Seven variables were considered to

of importance: AGE, LWT, RACE, SMOKE, HT, UI, and FTV. For comparison purposes, we begin to fit a logistic regression with conventional maximum-likelihood estimation:

```
data(lowbwt, package = "brinla")
lowbwt.glm1 <- glm(LOW ~ AGE + LWT + RACE + SMOKE + HT + UI + FTV,
    ↪ data=lowbwt, family=binomial())
round(coef(summary(lowbwt.glm1)), 4)
```

```
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.4548     1.1854  0.3837   0.7012
AGE          -0.0205     0.0360 -0.5703   0.5684
LWT          -0.0165     0.0069 -2.4089   0.0160
RACE2         1.2898     0.5276  2.4445   0.0145
RACE3         0.9191     0.4363  2.1065   0.0352
SMOKE1        1.0416     0.3955  2.6337   0.0084
HT1           1.8851     0.6948  2.7130   0.0067
UI1           0.9041     0.4486  2.0155   0.0439
FTV           0.0591     0.1720  0.3437   0.7311
```

In the output, the categorical variable RACE has been recoded as the two design variables, RACE2 and RACE3. In general, if a nominal scaled variable has k possible values, then $k-1$ design variables will be needed. Here RACE2 denotes the effect of a black mother relative to a white mother, and RACE3 denotes the effect of a mother in other races relative to a white mother. Similar recoding has been done for the variables, SMOKE, HT, UI. We then fit the logistic regression model using the INLA method:

```
lowbwt.inla1 <- inla(LOW ~ AGE + LWT + RACE + SMOKE + HT + UI + FTV,
    ↪ data=lowbwt, family = "binomial", Ntrials = 1, control.compute
    ↪ = list(dic = TRUE, cpo = TRUE))
round(lowbwt.inla1$summary.fixed, 4)
```

```
               mean     sd 0.025quant 0.5quant 0.975quant    mode kld
(Intercept)  0.5672 1.1853    -1.7289   0.5563     2.9236  0.5347   0
AGE         -0.0207 0.0360    -0.0921  -0.0204     0.0491 -0.0199   0
LWT         -0.0176 0.0069    -0.0317  -0.0174    -0.0047 -0.0169   0
RACE2        1.3405 0.5275     0.3151   1.3370     2.3851  1.3298   0
RACE3        0.9456 0.4362     0.1028   0.9409     1.8151  0.9314   0
SMOKE1       1.0749 0.3954     0.3140   1.0696     1.8664  1.0590   0
HT1          1.9727 0.6946     0.6595   1.9542     3.3909  1.9165   0
UI1          0.9331 0.4485     0.0524   0.9330     1.8130  0.9330   0
FTV          0.0559 0.1720    -0.2891   0.0585     0.3868  0.0635   0
```

We obtain estimates similar to those obtained when using the frequentist method. For example, the posterior mean of the parameter for LWT is -0.0176. Its estimated posterior standard deviation is 0.0069. The 2.5% and 97.5% posterior quantiles are both negative, which indicates with 95% probability that the effect for LWT is negative. Among all other predictors, the 95% credible intervals for AGE and FTV contain zero, while RACE2, RACE3, SMOKE1, HT1, and UI1 are positively associated with the outcome.

The odds ratios of the predictors can be calculated by exponentiating their estimated coefficients. For instance, the odds ratio for LWT is $\exp(-0.0176) = 0.9826$. It is interpreted as we expect to see 1.74% $(= 1 - 0.9826)$ decrease in the odds of

having a low birth weight baby for a one-unit increase in mother's weight, ass all other predictors are fixed.

We want to further obtain the reduced model while minimizing the numbe rameters. We may perform a backward elimination procedure using DIC (i.e., ally eliminating variables based on DIC; see the definition of DIC in Chapter 1 reduced model we obtain is the following:

```
lowbwt.inla2 <- inla(LOW ~ LWT + RACE + SMOKE + HT + UI, data=low
    ↪ family = "binomial", Ntrials = 1, control.compute = list(d
    ↪ TRUE, cpo = TRUE))
round(lowbwt.inla2$summary.fixed, 4)
```

```
              mean     sd 0.025quant 0.5quant 0.975quant    mode kld
(Intercept)  0.1087 0.9378    -1.6834   0.0915     2.0003  0.0567   0
LWT         -0.0175 0.0068    -0.0315  -0.0172    -0.0048 -0.0168   0
RACE2        1.3620 0.5214     0.3476   1.3587     2.3934  1.3522   0
RACE3        0.9486 0.4303     0.1198   0.9431     1.8091  0.9320   0
SMOKE1       1.0619 0.3925     0.3075   1.0563     1.8485  1.0451   0
HT1          1.9342 0.6907     0.6271   1.9163     3.3429  1.8799   0
UI1          0.9250 0.4475     0.0467   0.9249     1.8032  0.9246   0
```

In this reduced model, LWT has a negative effect with the estimated coeffi −0.0175; all other predictors have a positive effect on the regression coeffici We can compare the DICs for the full model and the reduced model:

```
c(lowbwt.inla1$dic$dic, lowbwt.inla2$dic$dic)
```

```
[1] 221.2093 217.7459
```

DIC for the reduced model is less than that for the full model, which indic that the reduced model has the better trade off between fit and model comple So, we prefer to use the reduced model. Its estimated logit is given by the follov expression:

$$\begin{aligned}\widehat{\text{logit}(\pi)} =& 0.109 - 0.018 \times \text{LWT} + 1.362 \times \text{RACE\_2} + 0.949 \times \text{RACE\_3} \\ &+ 1.062 \times \text{SMOKE} + 1.934 \times \text{HT} + 0.925 \times \text{UI},\end{aligned}$$

The equation can be used to obtain the fitted values, or make predictions for r observations.

4.3 Count Responses

In many application studies, the response variable of interest is the counted numb of occurrences of an event. In this type of data, the observations take only the nc negative integer values $\{0, 1, 2, 3, \ldots\}$, which arise from counting rather than rankir or grouping. The distribution of counts is discrete, and typically skewed. Applyi an ordinary linear regression model to these data could present at least two problen First, it is quite likely that the regression model will produce negative predicted va ues, which are theoretically impossible. Second, many distributions of count data a

positively skewed with many observations in the dataset having a value of 0. When one considers a transformation of the response variable (such as the log transformation, $\log(y+c)$, where c is a positive constant), the high number of 0's in the dataset prevents the transformation of a skewed distribution into normal.

4.3.1 Poisson Regression

The basic GLM for count data is the Poisson regression. Let Y be Poisson distributed with mean μ, where its probability mass function is $f(y|\mu) = \exp(-\mu)\mu^y/y!$. The distribution has the canonical parameter $\theta = \log\mu$, the dispersion parameter $\phi = 1$, and its variance function equals μ. The canonical link function, the logarithm link, leads to the Poisson regression model,

$$\begin{cases} Y_i \sim \text{Poisson}(\mu_i), \\ \log(\mu_i) = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}. \end{cases}$$

Let us consider a classical example for the simple Poisson regression. Whyte et al. (1987) reported the number of deaths due to AIDS in Australia per 3-month period from January 1983 to June 1986. The dataset only contains one predictor and one response with 14 observations, summarized in Table 4.3.

TABLE 4.3

Description for the variables in the `AIDS` data.

Variable Name	Description	Values
TIME	time measured in multiples of 3 months after January 1983	continuous
DEATHS	number of deaths in Australia due to AIDS	counts

Figure 4.1 displays the scatterplot of the data (the left panel) and the histogram for the response variable `DEATHS` (the right panel). We note that there is a nonlinear relationship between `TIME` and `DEATHS`, and `DEATHS` show a right-skewed distribution. A Poisson regression seems to be a reasonable choice to model the data. To fit the Poisson model using INLA, we use the following command:

```
AIDS.inla1 <- inla(DEATHS ~ TIME, data = AIDS, family = "poisson",
    ↪ control.compute = list(dic = TRUE, cpo = TRUE))
round(AIDS.inla1$summary.fixed, 4)
```

```
             mean     sd 0.025quant 0.5quant 0.975quant   mode kld
(Intercept) 0.3408 0.2512    -0.1690   0.3467     0.8183 0.3586   0
TIME        0.2565 0.0220     0.2142   0.2562     0.3008 0.2555   0
```

The coefficient table shows the posterior summary statistics for the unknown parameters in the model. We could write down the estimated equation for the mean response:

$$\hat{\mu} = \exp(0.3408 + 0.2565 \times \text{TIME}).$$

The `TIME` effect can be interpreted as follows: in the period between January 1983 to June 1986, the number of deaths due to AIDS in a year was on average $\exp(0.2565 \times$

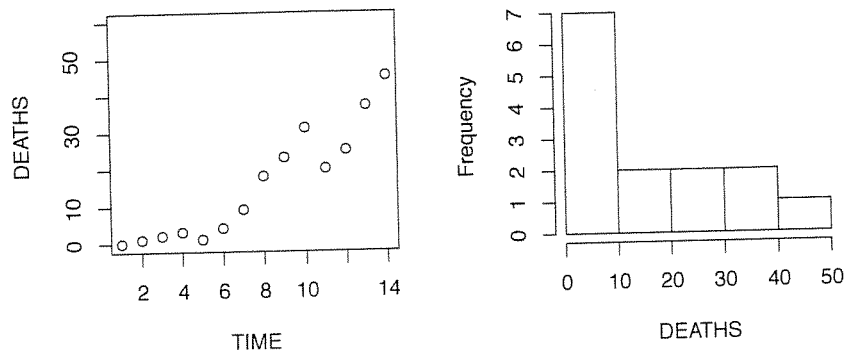

FIGURE 4.1
The scatterplot between TIME and DEATHS; and the histogram of DEATHS, in the AIDS data.

4) = 2.7899 times higher than in the year before. Next we generate the plot of the estimated mean function and its 95% credible interval:

```
plot(DEATHS ~ TIME, data=AIDS, ylim=c(0,60))
lines(AIDS$TIME, AIDS.inla$summary.fitted.values$mean, lwd=2)
lines(AIDS$TIME, AIDS.inla$summary.fitted.values$"0.025quant", lwd=1,
    ↪ lty=2)
lines(AIDS$TIME, AIDS.inla$summary.fitted.values$"0.975quant", lwd=1,
    ↪ lty=2)
```

From Figure 4.2, we note that in the beginning and the end of the time period the observed responses are less than the fitted values, while in the center period they are greater than the corresponding fitted values. This points to the fact that the model seems not entirely appropriate. Now let us consider log(TIME) instead of TIME as the explanatory variable. We fit the following model:

```
AIDS.inla2 <- inla(DEATHS ~ log(TIME), data=AIDS, family = "poisson",
    ↪ control.compute = list(dic = TRUE, cpo = TRUE))
round(AIDS.inla2$summary.fixed, 4)
```

```
               mean     sd 0.025quant 0.5quant 0.975quant    mode kld
(Intercept) -1.9424 0.5116    -2.9899  -1.9272    -0.9792 -1.8963   0
log(TIME)    2.1747 0.2150     1.7675   2.1692     2.6131  2.1581   0
```

The estimated equation is

$$\hat{\mu} = \exp(-1.9424 + 2.1747 \times \log(\text{TIME})).$$

The interpretation of this model is less intuitive: for a 1 unit increase in log(TIME the estimated count increases by a factor of exp(2.1747) = 8.7995. Let us look tl estimated mean function and its credible intervals for this model:

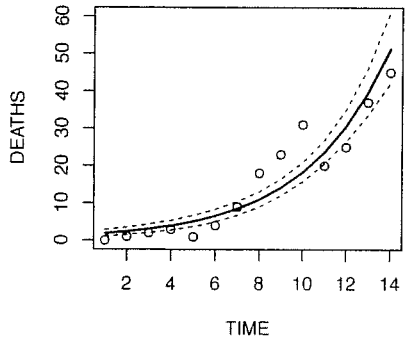

FIGURE 4.2
The estimated posterior mean function and its 95% credible interval for DEATHS when considering TIME as the explanatory variable.

```
plot(DEATHS ~ log(TIME), data = AIDS, ylim=c(0,60))
lines(log(AIDS$TIME), AIDS.inla2$summary.fitted.values$mean, lwd=2)
lines(log(AIDS$TIME), AIDS.inla2$summary.fitted.values$"0.025quant",
    ↪ lwd=1, lty=2)
lines(log(AIDS$TIME), AIDS.inla2$summary.fitted.values$"0.975quant",
    ↪ lwd=1, lty=2)
```

It appears that we obtain a better fit compared with the previous model. We could further check DICs for the two models:

```
c(AIDS.inla1$dic$dic, AIDS.inla2$dic$dic)
```

```
[1] 86.70308 74.10760
```

The second model has a smaller DIC, which confirms our findings from Figure 4.2 and Figure 4.3.

We may further compare the INLA results with the results using the conventional maximum likelihood estimation:

```
AIDS.glm <- glm(DEATHS ~ log(TIME), family=poisson(), data=AIDS)
round(coef(summary(AIDS.glm)), 4)
```

```
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.9442     0.5116 -3.8003    1e-04
log(TIME)     2.1748     0.2150 10.1130    0e+00
```

Very similar estimates are obtained from both methods.

4.3.2 Negative Binomial Regression

In applied work Poisson regression is restrictive in the analysis of count data. It is recognized that counts often display substantial extra-Poisson variation, or *overdis-*

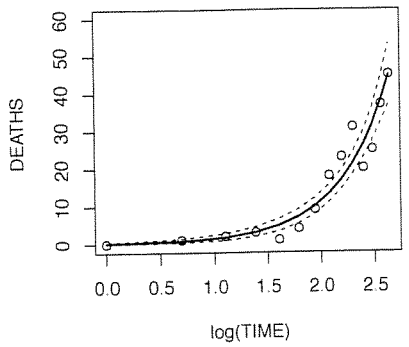

FIGURE 4.3
The estimated posterior mean function and its 95% credible interval for (DEATHS) when considering log(TIME) as the explanatory variable.

persion. Overdispersion refers to the situation when the variance of an observed dependent variable exceeds the nominal variance, given the respective assumed distribution. The assumption in Poisson model that the conditional mean and variance of Y given X are equal may be too strong and thus fail to account for the overdispersion. Inappropriate imposition of this restriction may result in unreasonably small estimated standard errors of the parameter estimates. Negative binomial regression is perhaps the most convenient way to relax the Poisson restriction and deal with the overdispersion.

More specifically, assume that $v_i, i = 1,...,n$ are unobserved random variables that follow a gamma distribution with shape parameter α and rate parameter α, Gamma(α,α); that is, $f(v) \propto x^{\alpha-1}\exp(-\alpha v)I\{v > 0\}$. Conditional on v_i, Y_i has a Poisson distribution with mean $v_i\mu_i$, i.e., $Y_i|v_i \sim$ Poisson$(v_i\mu_i)$. Then it follows that marginally Y_i has the negative binomial distribution given by

$$P(Y_i = y;\alpha,\mu_i) = \frac{\Gamma(y+\alpha)}{\Gamma(\alpha)y!}\left(\frac{\alpha}{\mu_i+\alpha}\right)^{\alpha}\left(\frac{\mu_i}{\mu_i+\alpha}\right)^{y}, \qquad (4.2)$$

where $y \in \{0,1,2,...\}$. The negative binomial distribution (4.2) will be denoted by $Y_i \sim NB(\alpha,\mu_i)$. It can be shown that the marginal mean and variance of Y_i are μ and $\mu_i + \mu_i^2/\alpha$, respectively. Thus, the parameter α quantifies the amount of overdispersion. Oftentimes, we define $\varphi = 1/\alpha$ as the dispersion parameter in the negative binomial model. The parameter $\varphi \to 0$ corresponds to no overdispersion. In such a case, the negative binomial model reduced to the Poisson model.

The log link is commonly used in negative binomial regression. Suppose we have a vector of p explanatory variables, $(x_{i1},...,x_{ip})$, that is related to the response Y_i. The

model is written as

$$\begin{cases} Y_i \sim \text{NB}(\alpha, \mu_i), \\ \log(\mu_i) = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}. \end{cases}$$

Let us use an example of nesting horseshoe crabs (Brockmann, 1996) to illustrate negative-binomial regression modeling. Agresti (2012) analyzed the data using the conventional frequentist GLM approach from Section 4.3 of his book. In this study, each female horseshoe crab in the study had a male crab attached to her in her nest. The study investigated factors that affect whether the female crab had any other males, called satellites, residing near her. Explanatory variables that are thought to affect this included the female crab's color (`COLOR`), spine condition (`SPINE`), weight (`WEIGHT`), and carapace width (`WIDTH`). The response variable for each female crab is her number of satellites (`SATELLITES`). The code sheet for the variables is displayed in Table 4.4. There are 173 females in this study.

TABLE 4.4
Code sheet for the variables in the `crab` data.

Variable Name	Description	Codes/Values
SATELLITES	the number of satellites for a female crab	counts
COLOR	crab's color	1 = light medium 2 = medium 3 = dark medium 4 = dark
SPINE	crab's spine condition	1 = both good 2 = one worn or broken 3 = both worn or broken
WEIGHT	crab's weight	kilogram (kg)
WIDTH	crab's carapace width	centimeter (cm)

It is always a good idea to start with descriptive statistics and plots. We first check the unconditional mean and variance of the outcome variable:

```
round(c(mean(crab$SATELLITES), var(crab$SATELLITES)), 4)
```

```
[1] 2.9191 9.9120
```

We note that the sample mean of `SATELLITES` is much lower than its variance. We further check the means and variances of `SATELLITES` by the crab's color type.

```
with(crab, tapply(SATELLITES, COLOR, function(x){round(mean(x), 4)}))
```

```
     1      2      3      4
4.0833 3.2947 2.2273 2.0455
```

```
with(crab, tapply(SATELLITES, COLOR, function(x){round(var(x), 4)}))
```

```
     1      2      3      4
 9.7197 10.2739  6.7378 13.0931
```

It seems that the variable `COLOR` is a good candidate for predicting `SATELLITES`,

since the mean value of the response appears to vary by COLOR. Also, we note that the variances within each level of COLOR are much higher than the means within each level. Let us plot the histogram and conditional histograms by COLOR for the response variable SATELLITES:

```
(p1 <- ggplot(crab, aes(x=SATELLITES)) + geom_histogram(binwidth=1,
    ↪ color="black"))
(p2 <- p1 + facet_wrap(~ COLOR, ncol=2))
```

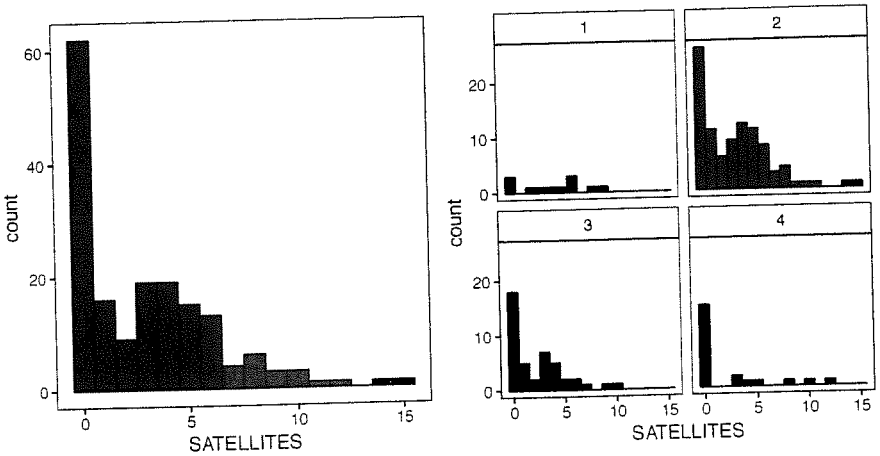

FIGURE 4.4
The histogram and conditional histograms (by COLOR) for the response variable SATELLITES, in the *crab* data.

Figure 4.4 shows the histogram of SATELLITES (the left panel) as well as the conditional histograms by crab's color type (the right panel). The histograms confirm our findings from summary statistics. These exploratory analysis results suggest that overdispersion is present and that a negative binomial model would be appropriate for the data.

By examining the correlation among the predictors, we notice that WEIGHT and WIDTH are highly correlated:

```
round(cor(crab$WEIGHT, crab$WIDTH),4)
```

```
[1] 0.8869
```

To avoid the multicollinearity problem, we only include the predictors COLOR, SPINE and WIDTH in the model. We first fit a negative binomial regression with convention maximum likelihood estimation:

```
library(MASS)
crab.glm <- glm.nb(SATELLITES ~ COLOR + SPINE + WIDTH, data=crab)
round(coef(summary(crab.glm)), 4)
```

```
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.3213     0.5637 -0.5700   0.5687
```

```
COLOR2      -0.3206     0.3725 -0.8607   0.3894
COLOR3      -0.5954     0.4159 -1.4317   0.1522
COLOR4      -0.5788     0.4643 -1.2467   0.2125
SPINE2      -0.2411     0.3934 -0.6130   0.5399
SPINE3       0.0425     0.2479  0.1713   0.8640
WIDTH        0.6925     0.1656  4.1826   0.0000
```

The following command fits the negative binomial regression with INLA:

```
crab.inla1 <- inla(SATELLITES ~ COLOR + SPINE + WIDTH, data = crab,
    ↪ family = "nbinomial", control.compute = list(dic = TRUE, cpo =
    ↪ TRUE))
round(crab.inla1$summary.fixed, 4)
```

```
              mean     sd 0.025quant 0.5quant 0.975quant    mode kld
(Intercept) -0.3158 0.5963    -1.4858  -0.3169     0.8592 -0.3187   0
COLOR2      -0.3216 0.3922    -1.1199  -0.3122     0.4244 -0.2941   0
COLOR3      -0.5988 0.4292    -1.4643  -0.5915     0.2257 -0.5775   0
COLOR4      -0.5814 0.4900    -1.5577  -0.5772     0.3708 -0.5691   0
SPINE2      -0.2467 0.3912    -1.0038  -0.2508     0.5346 -0.2587   0
SPINE3       0.0392 0.2527    -0.4661   0.0419     0.5287  0.0471   0
WIDTH        0.7001 0.1839     0.3457   0.6976     1.0691  0.6927   0
```

The results from the MLE approach and INLA are very close. We see that the posterior mean of the parameter for `WIDTH` is 0.7001 and its posterior standard deviation is 0.1839. Its 0.025 and 0.975 quantiles are both positive, which indicates with high probability that the effect for `WIDTH` is positive. The 95% credible intervals for all other predictors contain zero, so one cannot determine whether those effects are positive or negative based on the data.

In INLA, the size parameter α is represented as $\alpha = \exp(\theta)$ and a diffuse gamma distribution is defined on θ. By default, the summary of the posterior estimate of α is output:

```
round(crab.inla1$summary.hyperpar, 4)
```

```
                                                          mean         sd 0.025quant
size for the nbinomial observations (1/overdispersion) 0.9269     0.1572     0.6612
                                                       0.5quant 0.975quant       mode
size for the nbinomial observations (1/overdispersion)   0.915     1.2703      0.883
```

If we are interested in the overdispersion parameter, φ, the reciprocal of α, we could employ the function `inla.tmarginal`, which applies a transformation on the entire posterior distribution:

```
overdisp_post <- inla.tmarginal(fun=function(x) 1/x, marg=crab.inla1$
    ↪ marginals.hyperpar[[1]])
```

The posterior mean of φ can be obtained by the function `inla.emarginal`, which computes the expected value of a function `fun` applied to the marginal distribution `marg`:

```
round(inla.emarginal(fun=function(x) x, marg=overdisp_post), 4)
```

```
[1] 1.108
```

To obtain the posterior credible interval of φ, we apply the function `inla.qmarginal`:

```
round(inla.qmarginal(c(0.025, 0.975), overdisp_post), 4)
```

```
[1] 0.7610 1.5468
```

The above posterior summary of φ indicates a moderate overdispersion in the data set. We may want to compare the results using Poisson regression and negative binomial model:

```
crab.inla2 <- inla(SATELLITES ~ COLOR + SPINE + WIDTH, data = crab,
    ↪ family = "poisson", control.compute = list(dic = TRUE, cpo =
    ↪ TRUE))
round(crab.inla2$summary.fixed, 4)
```

```
              mean     sd 0.025quant 0.5quant 0.975quant    mode kld
(Intercept) -0.0491 0.2535    -0.5499  -0.0481     0.4457 -0.0461   0
COLOR2      -0.2695 0.1678    -0.5916  -0.2720     0.0673 -0.2770   0
COLOR3      -0.5232 0.1941    -0.9003  -0.5246    -0.1384 -0.5275   0
COLOR4      -0.5434 0.2253    -0.9866  -0.5430    -0.1023 -0.5423   0
SPINE2      -0.1615 0.2115    -0.5892  -0.1571     0.2420 -0.1483   0
SPINE3       0.0924 0.1195    -0.1393   0.0913     0.3297  0.0893   0
WIDTH        0.5475 0.0732     0.4028   0.5478     0.6901  0.5485   0
```

The posterior mean estimates of the predictors do not change too much, though the estimate of the intercept is very different. However, the posterior standard deviations from the Poisson regression are much less than those from the negative binomial regression. For over-dispersed data, Poisson regression underestimates the standard errors of the coefficients, leading to confidence intervals that are too narrow and potentially leading to incorrect inferences (Wang, 2012). We further compare the DICs for the negative binomial model and the Poisson model:

```
c(crab.inla1$dic$dic, crab.inla2$dic$dic)
```

```
[1] 761.2103 918.9907
```

The DIC of the negative binomial model is much smaller than that of the Poisson model, indicating that the negative binomial model is preferred in fitting the crab data.

4.4 Modeling Rates

In many applications, the count of an event is observed over a period or amount exposure, for example, traffic accidents per year, or count of deaths per age group. We often call the type of data, *rates*. A rate is a count of events divided by some measure of that unit's *exposure* (a particular unit of observation). Unlike a proportion, which ranges from 0 to 1, a rate could have any nonnegative value. Poisson negative-binomial regression are often appropriate for modeling rate data. In Poisson or negative-binomial model, this is handled as an *offset*, where the exposure variable enters on the right-hand side of the equation, but with a parameter estimate (log(exposure)) constrained to 1.

The following is a log-linked model for a rate as a function of the predictor va

ables, $(x_1, ..., x_p)$:

$$\log(\mu_i/e_i) = \beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip},$$

where μ_i is the mean event count and e_i is the exposure for the i^{th} observation. Note that the above equation can be rewritten as

$$\log(\mu_i) = \log(e_i) + \beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip}.$$

The model becomes a Poisson or negative binomial model in which the additional term on the right-hand side, $\log(e_i)$, is the offset, the log of the exposure.

Let us take a look at an example of car insurance claims (Aitkin et al., 2005). The data consist of the numbers of policyholders of an insurance company who were exposed to risk, and the numbers of car insurance claims made by those policyholders in the third quarter of 1973. The data include three four-level categorical predictors. The code sheet for the variables is displayed in Table 4.5.

TABLE 4.5

Code sheet for the variables in the insurance claim data.

Variable Name	Description	Codes/Values
District	the district of residence of policyholder	1 = rural; 2 = small towns; 3 = large towns; 4 = major cities.
Group	group of cars based on the engine capacity	<1 liter; 1–1.5 liter; 1.5–2 liter; >2 liter.
Age	age group of the policyholders	<25; 25–29; 30–35; > 35.
Holders	numbers of policyholders	counts
Claims	numbers of claims	counts

We want to model the relation between the rate of claims and the three explanatory variables, `District`, `Group`, and `Age`. We fit the rate data using Poisson regression with offset. Let us start the analysis with conventional maximum likelihood approach:

```
library(MASS)
data(Insurance, package = "MASS")
insur.glm <- glm(Claims ~ District + Group + Age + offset(log(Holders)
    ↪ ), data = Insurance, family = poisson)
round(summary(insur.glm)$coefficients, 4)
```

```
            Estimate Std. Error  z value Pr(>|z|)
(Intercept)  -1.8105     0.0330 -54.9102   0.0000
District2     0.0259     0.0430   0.6014   0.5476
District3     0.0385     0.0505   0.7627   0.4457
District4     0.2342     0.0617   3.7975   0.0001
```

```
Group.L      0.4297    0.0495   8.6881   0.0000
Group.Q      0.0046    0.0420   0.1103   0.9121
Group.C     -0.0293    0.0331  -0.8859   0.3757
Age.L       -0.3944    0.0494  -7.9838   0.0000
Age.Q       -0.0004    0.0489  -0.0073   0.9942
Age.C       -0.0167    0.0485  -0.3452   0.7299
```

Note that, to specify the model correctly in `glm` function, we must include the term log(Holders) as an explanatory variable with a coefficient of 1. That is, log(Holders) is taken as an offset for the model by specifying "`offset(log(Holders))`" in the model formula.

When we fit a Bayesian model using `inla` function, the offset term needs to be specified by the argument `E = Holders`:

```
insur.inla1 <- inla(Claims ~ District + Group + Age, data = Insurance,
    ↪ family = "poisson", E = Holders)
round(insur.inla1$summary.fixed, 4)
```

```
               mean     sd 0.025quant 0.5quant 0.975quant    mode kld
(Intercept) -1.8122 0.0330    -1.8774  -1.8120    -1.7479 -1.8117   0
District2    0.0259 0.0430    -0.0588   0.0259     0.1101  0.0261   0
District3    0.0385 0.0505    -0.0612   0.0387     0.1372  0.0391   0
District4    0.2342 0.0617     0.1118   0.2346     0.3541  0.2355   0
Group.L      0.4296 0.0495     0.3320   0.4298     0.5262  0.4301   0
Group.Q      0.0043 0.0420    -0.0787   0.0044     0.0862  0.0047   0
Group.C     -0.0294 0.0331    -0.0943  -0.0294     0.0356 -0.0295   0
Age.L       -0.3943 0.0494    -0.4900  -0.3947    -0.2961 -0.3956   0
Age.Q       -0.0002 0.0489    -0.0964  -0.0001     0.0956  0.0000   0
Age.C       -0.0164 0.0485    -0.1116  -0.0164     0.0787 -0.0163   0
```

From the output above, we note that the effect of major cities, `District4`, is 0.2342 with the 95% credible interval (0.1118,0.3541). The results can be interpreted as follows: the estimated rate of claims for major cities is 26.36% = exp(0.234) − exp(0) with credible levels (11.83%, 42.49%) = (exp(0.1118) − exp(0), exp(0.3541) − exp(0)), higher than that of claims for rural areas, assuming the group and age effects are fixed. Similar statements can be made for the two other significant effects `Group.L` and `Age.L`.

To check the validation of a Poisson model, Pearson residuals are often used for model diagnostics in frequentist analysis,

$$\hat{\varepsilon}_i = (y_i - \hat{\mu}_i)/\sqrt{\hat{\mu}_i}, i = 1, ..., n,$$

where $\hat{\mu}_i$ is the maximum likelihood estimate. The residuals approximately follow standard normal distribution if the model is correctly specified. By analogy with classical model checking, we define Bayesian Pearson residuals as

$$r_i = (y_i - \mu_i)/\sqrt{\mu_i} = \frac{y_i - g^{-1}(\mathbf{x}_i, \boldsymbol{\beta})}{\sqrt{g^{-1}(\mathbf{x}_i, \boldsymbol{\beta})}}, i = 1, ..., n.$$

Each r_i is just a function of unknown parameters, and its posterior distribution is straightforward to calculate (see more discussions regarding Bayesian residuals in Chapter 3). Plotting the posterior mean or median of the r_i's versus the index of the observations or the fitted values might reveal failure in model assumptions.

We have written a convenience function, `bri.Pois.resid`, to calculate Bayesian Pearson residuals (posterior means) for Poisson regression in our `brinla` library. When the argument `plot = TRUE` in the function is specified, a residual plot by case is output:

```
insur.bresid <- bri.Pois.resid(insur.inla1, plot = TRUE)
abline(1.96, 0, lty=2); abline(-1.96, 0, lty=2)
qqnorm(insur.bresid$resid); qqline(insur.bresid$resid)
```

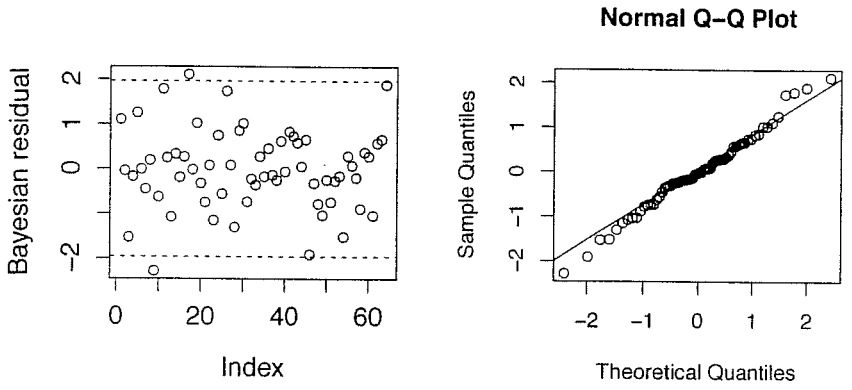

FIGURE 4.5
Bayesian Pearson residual plots of Poisson model for the insurance claim data. The left panel is the plot of residuals by case, and the right panel is the normal QQ plot for the residuals.

The left panel in Figure 4.5 shows the plot of residuals by case (index plot). We observe a horizontal band with points that vary at random. The right panel in Figure 4.5 shows a normal QQ plot for the residuals, indicating the residuals fit to a standard normal distribution very well. Thus, the assumption of a Poisson model for the insurance claim data is well supported.

4.5 Gamma Regression for Skewed Data

Poisson and negative binomial models are very popular in practice, but there are a number of other GLMs which are useful for particular types of data. The gamma GLM can be used for continuous but skewed responses. The most common way to analyze such data is to log transform the responses. However, modeling the skewed data with gamma distribution in GLM framework may give better interpretability, since gamma regression parameters are interpretable in terms of the mean of the