CHAPTER

# 8

# Normal Linear Regression

OUTLINE



## 8.1 INTRODUCTION

We have seen in Chapter 6 that the linear model underlying the simple normal linear regression is the same as that for the t-test:

$$y_i = \alpha + \beta * x_i + \varepsilon_i$$
$$\varepsilon_i \sim Normal\,(0, \sigma^2)$$

The only difference is that the variable $x_i$ doesn't just take on two possible values to indicate membership to one of two groups; rather, it is a measurement that can take on any possible value, within some bounds and up to measurement accuracy. The geometric representation of this model is a straight line, with $\alpha$ being the intercept and $\beta$ the slope.

As a motivating example for a linear regression analysis, we take a Swiss survey of the wallcreeper (Fig. 8.1), a spectacular little cliff-inhabiting bird that appears to have declined greatly in Switzerland in recent years.


*Introduction to WinBUGS for Ecologists*
DOI: 10.1016/B978-0-12-378605-0.00008-9

FIGURE 8.1 Wallcreeper (*Tichodroma muraria*), Switzerland, 1989. (*Photo E. Hüttenmoser*)

Assume that we had data on the proportion of sample quadrats in which the species was observed in Switzerland for the years 1990–2005 and that we were willing to assume that the random deviations about a linear time trend were normally distributed. This is for illustration only; usually, we would use logistic regression (Chapters 17–19) or a site-occupancy model (see Chapter 20) to make inference about such data that have to do with the distribution of a species and represent a proportion (i.e., number sites occupied/number sites surveyed).

Importantly, in this chapter, we will also introduce posterior predictive model checking, including the Bayesian *p*-value (Gelman et al., 1996; Gelman and Hill, 2007, Chapter 24). This is a very general concept for checking the goodness-of-fit of a model analysed using simulation techniques like MCMC.

## 8.2 DATA GENERATION

We generate simple linear regression data (see later Fig. 8.4):

```
n <- 16                          # Number of years
a = 40                           # Intercept
b = -1.5                         # Slope
sigma2 = 25                      # Residual variance
```

```
x <- 1:16                              # Values of covariate year
eps <- rnorm(n, mean = 0, sd = sqrt(sigma2))
y <- a + b*x + eps                     # Assemble data set
plot((x+1989), y, xlab = "Year", las = 1, ylab = "Prop. occupied (%)", cex = 1.2)
```

## 8.3 ANALYSIS USING R

Here is a classical analysis using the linear regression facility in R. The function `I(.)` makes it more straightforward to plot the results:

```
print(summary(lm(y ~ I(x+1989))))
abline(lm(y~ I(x+1989)), col = "blue", lwd = 2)
```

## 8.4 ANALYSIS USING WinBUGS

Next, we conduct a Bayesian analysis of the same model, which will also include a posterior predictive check plus a Bayesian *p*-value (Gelman et al., 1996) to assess the adequacy of the model for our data set. (Hint: If you don't understand some WinBUGS expressions, such as `pow()` or `step()`, open the WinBUGS manual under the Help menu, and in the contents go to Model Specification > Logical nodes.)

### 8.4.1 Fitting the Model

```
# Write model
sink("linreg.txt")
cat("
model {

# Priors
 alpha ~ dnorm(0,0.001)
 beta ~ dnorm(0,0.001)
 sigma ~ dunif(0, 100)

# Likelihood
 for (i in 1:n) {
    y[i] ~ dnorm(mu[i], tau)
    mu[i] <- alpha + beta*x[i]
 }

# Derived quantities
 tau <- 1/ (sigma * sigma)
 p.decline <- 1-step(beta)       # Probability of decline
```

```
# Assess model fit using a sums-of-squares-type discrepancy
 for (i in 1:n) {
     residual[i] <- y[i]-mu[i]          # Residuals for observed data
     predicted[i] <- mu[i]              # Predicted values
     sq[i] <- pow(residual[i], 2)       # Squared residuals for observed data

# Generate replicate data and compute fit stats for them
     y.new[i] ~ dnorm(mu[i], tau) # one new data set at each MCMC iteration
     sq.new[i] <- pow(y.new[i]-predicted[i], 2) # Squared residuals for new data
 }
 fit <- sum(sq[])                   # Sum of squared residuals for actual data set
 fit.new <- sum(sq.new[])           # Sum of squared residuals for new data set
 test <- step(fit.new - fit)        # Test whether new data set more extreme
 bpvalue <- mean(test)              # Bayesian p-value
}
",fill=TRUE)
sink()

# Bundle data
win.data <- list("x","y", "n")

# Inits function
inits <- function(){ list(alpha=rnorm(1), beta=rnorm(1), sigma = rlnorm(1))}

# Parameters to estimate
params <- c("alpha","beta", "p.decline", "sigma", "fit", "fit.new", "bpvalue",
"residual", "predicted")

# MCMC settings
nc = 3 ; ni=1200 ; nb=200 ; nt=1

# Start Gibbs sampler
out <- bugs(data = win.data, inits = inits, parameters = params, model =
"linreg.txt", n.thin = nt, n.chains = nc, n.burnin = nb, n.iter = ni, debug = TRUE)

print(out, dig = 3)
```

## 8.4.2 Goodness-of-Fit Assessment in Bayesian Analyses

In the WinBUGS code, there are two components included to assess the goodness-of-fit of our model. First, there are two lines that compute residuals and predicted values under the model. And second, there is code to compute a Bayesian *p*-value, i.e., a posterior predictive check (Gelman et al., 1996, 2004; Gelman and Hill, 2007). As an instructive example, we will assess the adequacy of the model using a traditional residual check and then using posterior predictive distributions, including a Bayesian *p*-value, as an overall measure of fit for a chosen fit criterion.

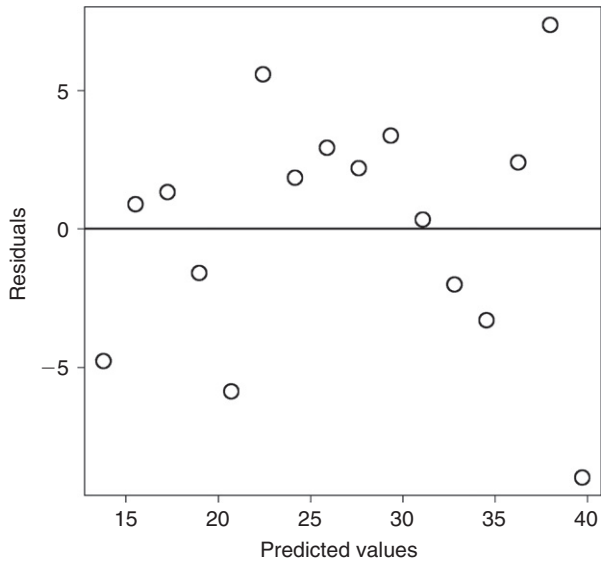


FIGURE 8.2 Residual plot for the linear regression analysis for trend in the Swiss wallcreeper distribution.

### Residual Plots

One commonly produced graphical check of the residuals of a linear model is a plot of the residuals against the predicted values. Under the normal linear regression model, residuals are assumed to be a random sample from one single normal distribution. There should be no visible structure in the residuals. In particular, the scatterplot of the residuals should not have the shape of a fan which would indicate that the variance is not constant but is larger, or smaller, for larger responses. We check this first and find no sign of a violation of the homoscedasticity assumption (Fig. 8.2).

```
plot(out$mean$predicted, out$mean$residual, main = "Residuals vs. predicted
values", las = 1, xlab = "Predicted values", ylab = "Residuals")
abline(h = 0)
```

### Posterior Predictive Distributions and Bayesian p-Values

The use of posterior predictive distributions is a very general way of assessing the fit of a model when using MCMC model fitting techniques (Gelman et al., 1996; Gelman and Hill, 2007). The idea of a posterior predictive check is to compare the lack of fit of the model for the actual data set with the lack of fit of the model when fitted to replicated, "ideal" data sets. Ideal means that a data set conforms exactly to the assumptions made by the model and is generated under the parameter estimates obtained

from the analysis of the actual data set. In contrast to a frequentist analysis, where the solution of a model consists in a single value for each parameter, we estimate a distribution in a Bayesian analysis; hence, any lack-of-fit statistic will also have a distribution.

To obtain such perfect data sets, at each MCMC iteration one replicate data set is assembled under the same model that we fit to the actual data set and using the values of all parameters from the current MCMC iteration. A discrepancy measure chosen to embody a certain kind of lack of fit is computed for both that perfect data set and for the actual data set. Therefore, at the end of an MCMC run for $n$ chains of length $m$, we have $n*m$ draws from the posterior predictive distribution of the discrepancy measure applied to the actual data set as well as for the discrepancy measure applied to a perfect data set.

What does "discrepancy measure" mean and how is it chosen? The discrepancy measure can be chosen to assess particular features of the model. Often, some global measure of lack of fit will be selected, e.g., a sums of squares-type of discrepancy as we do here, or a Chi-squared-type discrepancy (see Chapter 21 for another example in a more complex hierarchical model). However, entirely different measures may also be chosen; for instance, a discrepancy measure that quantifies the incidence or magnitude of extreme values to assess the adequacy of the model for outliers; see Gelman et al. (1996) for examples.

One of the best ways to assess model adequacy based on posterior predictive distributions is graphically, in a plot of the lack of fit for the ideal data vs. the lack of fit for the actual data (Fig. 8.3). If the model fits the data, then about half of the points should lie above and half of them below a 1:1 line. Alternatively, a numerical summary, called a Bayesian $p$-value, can be computed that quantifies the proportion of times when the discrepancy measure for the perfect data sets is greater than the discrepancy measure computed for the actual data set. A fitting model has a Bayesian $p$-value near 0.5, and values close to 0 or close to 1 suggest doubtful fit of the model.

```
lim <- c(0, 3200)
plot(out$sims.list$fit, out$sims.list$fit.new, main = "Graphical posterior
predictive check", las = 1, xlab = "SSQ for actual data set", ylab = "SSQ for ideal
(new) data sets", xlim = lim, ylim = lim)
abline(0, 1)

mean(out$sims.list$fit.new > out$sims.list$fit) # Bayesian p-value
> mean(out$sims.list$fit.new > out$sims.list$fit)
[1] 0.547
```

The graphical posterior predictive check and the numerical Bayesian $p$-value concur in suggesting that our fitted model is adequate for the

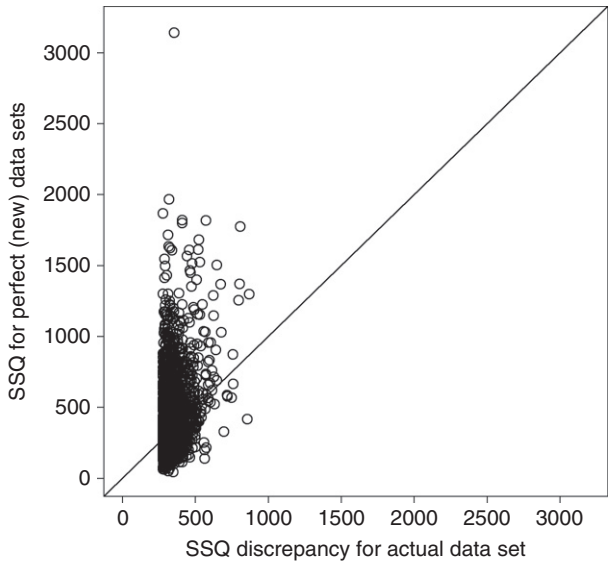


FIGURE 8.3 Graphical posterior predictive check (PPC) of the model adequacy for the wallcreeper analysis plotting predictive vs. realized sums-of-squares discrepancies. The Bayesian *p*-value is equal to the proportion of plot symbols above the 1:1 line. Note that the truncation on the left is due to the hard minimum provided by the least squares estimate.

wallcreeper data, something that will hardly come as a surprise with these simulated data.

Some statisticians don't like posterior predictive checks because they use the data twice: first, to generate the replicate data and second to compare them with these replicates. Model fit assessments based on posterior predictive checks are somewhat too liberal, and posterior predictive checks should not be used for model selection; see Chapter 10 in Ntzoufras (2009) for alternatives.

### 8.4.3 Forming Predictions

Predictions are expected values of the response variable at some hypothetical values of one or more explanatory variables. Forming predictions is extremely important in applied statistical modeling for two reasons. First, predictions, especially when represented as a graph, are one of the best ways of communicating what can be learned from a model. Second, especially for more complex models, for instance, when there are polynomial terms or interactions and also for Poisson or binomial models (see Chapters 13–21), predictions may be the only way to understand what a model is telling us. Of course, for the simple normal straight-line

model in this chapter, we can simply look at magnitude and sign of the slope estimate to understand what the model is telling us about the population trend in Swiss wallcreepers. However, as an exercise, we next plot the estimated trend line from the classical (maximum likelihood [ML]) and the Bayesian (MCMC) fit of the linear regression model:

```
plot((x+1989), y, xlab = "Year", las = 1, ylab = "Prop. occupied (%)", cex = 1.2)
abline(lm(y~ I(x+1989)), col = "blue", lwd = 2)
pred.y <- out$mean$alpha + out$mean$beta * x
points(1990:2005, pred.y, type = "l", col = "red", lwd = 2)
text(1994, 20, labels = "blue - ML; red - MCMC", cex = 1.2)
```

Given the small sample size, we get remarkably similar and indeed virtually identical inferences under the two paradigms (Fig. 8.4).

We can also easily calculate a 95% uncertainty interval by simulation. There are two kinds of such an uncertainty interval, one for the actual data set (called a credible interval) and another for a new data set sampled from the same population (called a prediction interval). Because there is more uncertainty about the line when adding the variability due to the new sample, the latter is wider than the former. Here, we give an example of a credible interval. To produce the interval, we compute the expected response for each of the 3000 elements of our posterior sample of the intercept $\alpha$ and the slope $\beta$, and at each point along the $x$-axis (i.e., for each of

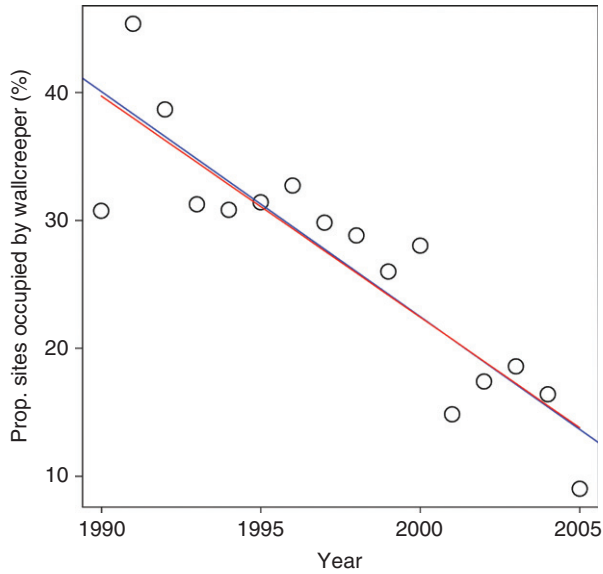


FIGURE 8.4 Observed and predicted change in the distribution of Swiss wallcreepers (blue – maximum likelihood, red – Bayesian).

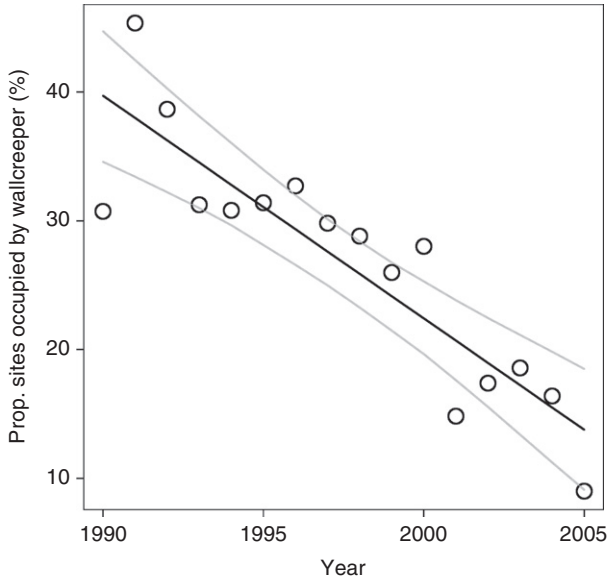


FIGURE 8.5 Predicted wallcreeper trend (black line) with 95% credible interval (grey lines).

the 16 years). Then, we use the 2.5th and 97.5th percentiles of each posterior distribution as our bounds of the credible interval.

We set up an R data structure to hold the predictions, fill them, then determine the appropriate percentile points and produce a plot (Fig. 8.5):

```
predictions <- array(dim = c(length(x), length(out$sims.list$alpha)))
for(i in 1:length(x)){
    predictions[i,] <- out$sims.list$alpha + out$sims.list$beta*i
}
LPB <- apply(predictions, 1, quantile, probs = 0.025) # Lower bound
UPB <- apply(predictions, 1, quantile, probs = 0.975) # Upper bound

plot((x+1989), y, xlab = "Year", las = 1, ylab = "Prop. occupied (%)", cex = 1.2)
points(1990:2005, out$mean$alpha + out$mean$beta * x, type = "l", col = "black",
lwd = 2)
points(1990:2005, LPB, type = "l", col = "grey", lwd = 2)
points(1990:2005, UPB, type = "l", col = "grey", lwd = 2)
```

## 8.4.4 Interpretation of Confidence vs. Credible Intervals

Consider the frequentist inference about the slope parameter; –1.763, SE 0.241. A quick and dirty frequentist 95% confidence interval is

provided by −1.763 ± 2*0.241 = (−2.245, −1.281). This means that if we took, for example, 100 replicate sample observations of 16 annual surveys each in the same Swiss wallcreeper population and 100 times estimated an annual trend with associated 95% CI using linear regression, then on average we would expect 95 intervals would indeed contain the true value of the population trend. We cannot make any direct probability statement about the trend itself; the true value of the trend is either in or out of our single interval, but there is no probability associated with this. In particular, it is wrong to say that the population trend of the wallcreeper lies between −2.245 and −1.281 with a probability of 95%. The probability statement in the 95% CI refers to the reliability of the tool, i.e., computation of the confidence interval, and not to the parameter for which a CI is constructed.

In contrast, the posterior probability in a Bayesian analysis measures our degree of belief about the likely magnitude of a parameter, given the model, the observed data, and our priors. Hence, we can make direct probability statements about a parameter using its posterior distribution. Let's do this here for the slope parameter, which represents the population trend of the wallcreeper in Switzerland (Fig. 8.6).

```
hist(out$sims.list$beta, main = "", col = "grey", xlab = "Trend estimate", xlim =
c(-4, 0))
abline(v = 0, col = "black", lwd = 2)
```

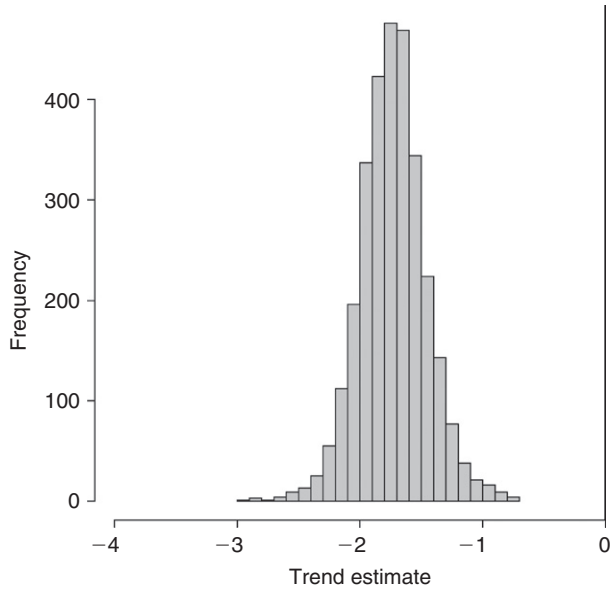


FIGURE 8.6 Posterior distribution of the distributional trend in Swiss wallcreepers. The value of zero (representing no trend) is shown as a black vertical line.

We see clearly that values representing no decline or an increase, i.e., values of the slope of 0 and larger have no mass at all under this posterior distribution. We can thus say that the probability of a stable or increasing wallcreeper population is essentially nil. Such a statement is exactly what most users of statistics, such as politicians, would like to have, rather than the somewhat contorted statement about a population trend as based on the frequentist confidence interval.

## 8.5 SUMMARY

We have used WinBUGS to fit a linear regression, the algebraic model and the WinBUGS code for which is essentially identical to that for the t-test. We have also introduced posterior predictive distributions along with the Bayesian *p*-value as a very general and flexible way of assessing goodness-of-fit of a model analyzed using MCMC.

### EXERCISES

1. *Toy problem*: Assume you examined five frogs that weighed 10, 20, 23, 32, and 35 g and had lengths of 5, 7, 10, 12, and 15 units. Write out the linear regression model using vectors and matrices and the set of equations implied. Conduct a normal linear regression analysis for this data set using R and WinBUGS.
2. *Prediction*: One way of forming predictions in WinBUGS is by specifying them as additional (derived) variables in the model. Another way is to form them outside of WinBUGS in R, if Markov chains for all the required ingredients are available. The third and perhaps the simplest way is by adding to the data set missing values (NAs) in the response and the desired levels of the explanatory variables in the model. In a Bayesian analysis, missing values are treated exactly like parameters, i.e., WinBUGS will draw samples for each missing value as part of the model fitting. The resulting posterior predictive distribution can then be summarized for inference in the usual way. Try this for the frog data set and predict mass at length 16, 17, 18, 19, and 20 units.
3. *Swiss hare data*: Fit a normal linear regression analysis for *mean.density* on *year* for grassland areas. Hint: You must first select the data for grassland areas and then aggregate over sites to obtain mean annual density in grassland. Does a linear trend adequately capture the variability in the data?