PART TWO

Models and Methods

7

FREQUENCY MODELS

"Statistics is the grammar of science."
—Karl Pearson

Here in Part II, we focus on statistical models for understanding and predicting hurricane climate. This chapter shows you how to model hurricane occurrence. This is done using the annual count of hurricanes making landfall in the United States. We also consider the occurrence of hurricanes across the basin and by origin.

We begin with exploratory analysis and then show you how to model counts with Poisson regression. Issues of model fit, interpretation, and prediction are considered in turn. The topic of how to assess forecast skill is examined including how to perform cross-validation. Alternatives to the Poisson regression model are considered. Logistic regression and receiver operating characteristics (ROCS) are also covered.

7.1 COUNTS

You use the data set *US.txt* which contains a list of tropical cyclone counts by year (see Chapter 2). The counts indicate the number of hurricanes hitting in the United States (excluding Hawaii). Input the data, save them as a data frame object, and print out the first six lines by typing

```
> H = read.table("US.txt", header=TRUE)
> head(H)
  Year All MUS G FL E
1 1851   1   1 0  1 0
2 1852   3   1 1  2 0
3 1853   0   0 0  0 0
4 1854   2   1 1  0 1
```

```
5 1855   1   1 1  0 0
6 1856   2   1 1  1 0
```

The columns include year `Year`, number of U.S. hurricanes `All`, number of major U.S. hurricanes `MUS`, number of U.S. Gulf coast hurricanes `G`, number of Florida hurricanes `FL`, and number of East coast hurricanes `E`. Save the number of years in the record as `n` and the average number hurricanes per year as `rate`.

```
> n = length(H$Year); rate = mean(H$All)
> n; rate
[1] 160
[1] 1.69
```

The average number of U.S. hurricanes is 1.69 per year over these 160 years.

First plot a time series and a distribution of the annual counts. Together, the two plots provide a nice summary of the information in your data relevant to any modeling effort.

```
> par(las=1)
> layout(matrix(c(1, 2), 1, 2, byrow=TRUE),
+    widths=c(3/5, 2/5))
> plot(H$Year, H$All, type="h", xlab="Year",
+    ylab="Hurricane Count")
> grid()
> mtext("a", side=3, line=1, adj=0, cex=1.1)
> barplot(table(H$All), xlab="Hurricane Count",
+    ylab="Number of Years", main="")
> mtext("b", side=3, line=1, adj=0, cex=1.1)
```

The `layout` function divides the plot page into rows and columns as specified in the matrix function (first argument). The column widths are specified using the `width` argument. The plot symbol is a vertical bar (`type="h"`). The tic labels on the vertical axis are presented in whole numbers consistent with count data.

Figure 7.1 shows the time series and distribution of annual hurricanes over the 160-year period. There is a total of 271 hurricanes. The year-to-year variability and the distribution of counts appear to be consistent with a random count process. There are 34 years without a hurricane and one year (1886) with seven hurricanes. The number of years with a particular hurricane count provides a histogram. It is a good research practice to show your data in t his way.

7.1.1 Poisson Process

The shape of the histogram suggests that a Poisson distribution is a good description for these data. The density function of the Poisson distribution shows that the

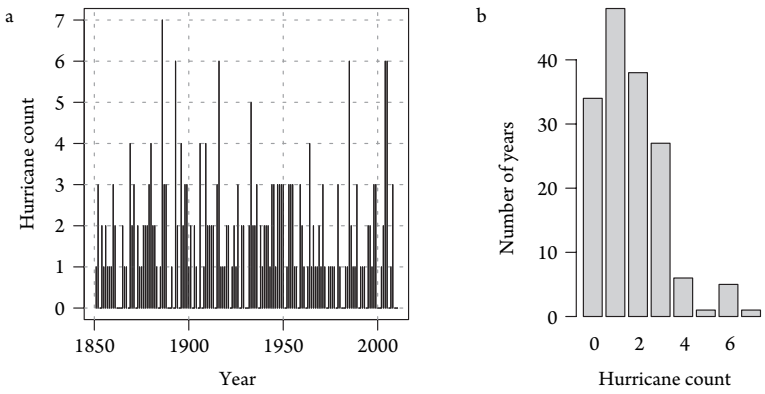

Figure 7.1 Annual hurricane occurrence. (a) Time series and (b) distribution.

probability p of obtaining a count x when the mean count (rate) is λ is given by

$$p(x) = \frac{e^{-\lambda}\lambda^x}{x!} \tag{7.1}$$

where e is the exponential function and ! is the factorial symbol. The equation indicates that probability of no events is $p(0) = e^{-\lambda}$.

With λ =1.69 hurricanes per year, the probability of no hurricanes in a random year is

```
> exp(-rate)
[1] 0.184
```

This implies that the probability of at least one hurricane is 0.82 or 82 percent.

Using the `dpois` function, you can determine the probability for any number of hurricanes. For example, to determine the probability of observing exactly one hurricane when the rate is 1.69 hurricanes per year, type

```
> dpois(x=1, lambda=rate)
[1] 0.311
```

Or the probability of five hurricanes expressed as a percentage is

```
> dpois(5, rate) * 100
[1] 2.14
```

Recall that you can leave off the argument names in the function if the argument values are placed in the correct order. Remember, the argument order can be found by placing a question mark in front of the function name and leaving off the parentheses. This brings up the function's help page (see Chapter 2).

To answer the question, *What is the probability of two or fewer hurricanes?* you use the cumulative probability function `ppois` as follows:

```
> ppois(q=2, lambda=rate)
[1] 0.759
```

Then to answer the question, *What is the probability of more than two hurricanes?* you add the argument `lower.tail=FALSE`.

```
> ppois(q=2, lambda=rate, lower.tail=FALSE)
[1] 0.241
```

7.1.2 Inhomogeneous Poisson Process

The Poisson distribution has the property that the variance is equal to the mean. Thus data that can be described with a Poisson distribution has a variance to mean ratio close to one. You compute this ratio with your data by typing

```
> round(var(H$All)/rate, 2)
[1] 1.24
```

This says that the variance of hurricane counts is 24 percent larger than the mean. Is this unusual for a Poisson distribution?

You check by performing a Monte Carlo (MC) simulation experiment. An MC simulation relies on repeated random sampling from a distribution. Compare it to a bootstrap resampling procedure, which relies on repeated random sampling from a set of data (see Section 3.9.3). A single random sample of size n from a Poisson distribution with a rate equal to 1.5 is obtained by typing

```
> rpois(n=5, lambda=1.5)
[1] 0 1 0 2 1
```

Here you repeat this $m = 1{,}000$ times and let n be the number of years in your hurricane record and λ be the rate. For each sample, you compute the ratio of the variance to the mean.

```
> set.seed(3042)
> ratio = numeric()
> m = 1000
> for (i in 1:m){
+ h = rpois(n=n, lambda=rate)
+ ratio[i] = var(h)/mean(h)
+ }
```

The vector `ratio` contains 1,000 values of the ratio. To help answer the *Is this unusual?* question, you determine the proportion of ratios greater than 1.24:

```
> sum(ratio > var(H$All)/rate)/m
[1] 0.028
```

Only 2.8 percent of the ratios are larger, so the answer from your MC experiment is "yes," and variability in hurricane counts is higher than you would expect from a Poisson distribution with a constant rate.

This might indicate that the rate varies with time. Although you can compute a long-term average, some years have a higher rate than others. The variation in the rate is due to things such as El Niño. So you expect more variance (extra dispersion) in counts relative to a constant rate (homogeneous Poisson) distribution. This is the rationale behind seasonal forecasts. The variation in the annual rate is not obvious from looking at the variation in counts. Even with a constant rate, the counts will vary.

You modify your MC simulation using the gamma distribution for the rate and then examine the ratio of variance to the mean from a set of Poisson counts with the variable rate. The gamma distribution describes the variability in the rate using the shape and scale parameters. The mean of the gamma distribution is the shape times the scale. You specify the shape to be 5.6 and the scale to be 0.3 so that the product matches closely the long-term average count. You could choose other values that produce the same average.

Now your simulation first generates 1,000 random gamma values, and then for each gamma, 160 years of hurricane counts are generated.

```
> ratio = numeric(); set.seed(3042); m = 1000
> for (i in 1:m){
+   h = rpois(n=n, lambda=rgamma(m, shape=5.6,
+   scale=.3))
+   ratio[i] = var(h)/mean(h)
+ }
> sum(ratio > var(H$All)/rate)/m
[1] 0.616
```

In this case, we find that 61.6 percent of the ratios are larger, so we conclude that the observed hurricane counts are more consistent with a variable-rate (inhomogeneous) Poisson model.

These examples demonstrate an important use of statistics: to simulate data that have the same characteristics as your observations. Figure 7.2 shows a plot of the observed hurricane record over the 160-year period together with plots from three simulated records of the same length and having the same overdispersed Poisson variation as the observed record. As shown earlier, such simulated records provide a way to test hypotheses about natural variability in hurricane climate. Summary characteristics of a 100 years of hurricanes at a coastal location may be of little value, but running a sediment transport model at that location with a much larger number of simulated hurricane counts will provide an assessment of the uncertainty in sediment movement resulting from variation in hurricane frequency.

7.2 ENVIRONMENTAL VARIABLES

The parameter of interest is the annual hurricane rate. Given the rate, you have a probability distribution for any possible hurricane count. You noted that the observed

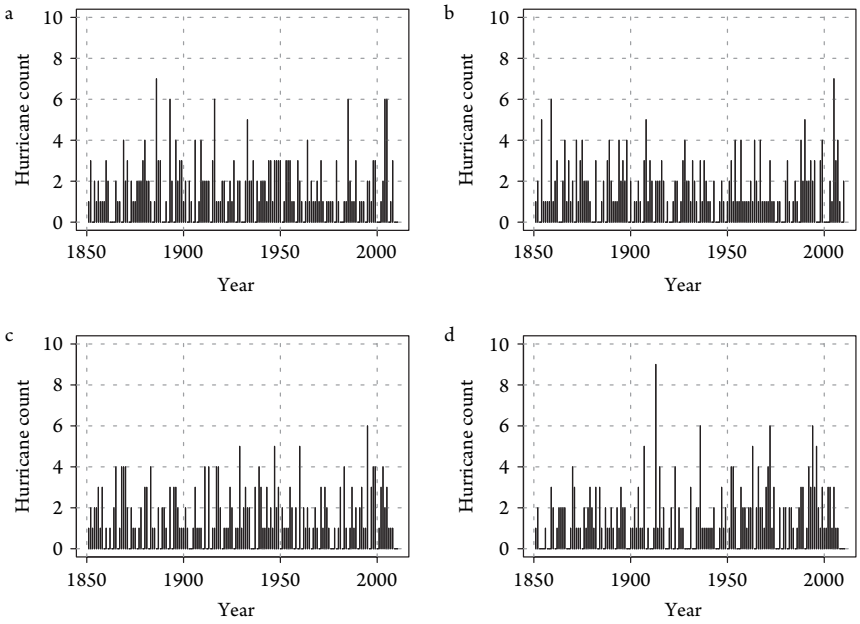

Figure 7.2 Hurricane occurrence using (a) observed and (b–d) simulated counts.

counts are consistent with a Poisson distribution having a variable rate. But where does this variability come from?

On the annual time scale to a first-order [illegible] ocean heat content and cold upper air temperature provide the fuel for a hurricane, a calm atmosphere allows a hurricane to intensify, and the position and strength of the subtropical high pressure steers a hurricane that does form. Thus, hurricane activity responds to changes in climate conditions including sea-surface temperature (SST) as an indicator of oceanic heat content, sunspot number as an indicator of upper air temperature, El Niño-Southern Oscillation (ENSO) as indicator of wind shear, and the North Atlantic Oscillation as an indicator of steering flow.

SST provides an indication of the thermodynamic environment, as do sunspots. An increase in solar UV radiation during periods of strong solar activity will have a suppressing effect on tropical cyclone intensity as the air above the hurricane will warm through absorption of radiation by ozone. ENSO is characterized by basin-scale fluctuations in sea-level pressure (SLP) across the equatorial Pacific Ocean. The Southern Oscillation Index (SOI) is defined as the normalized SLP difference between Tahiti and Darwin. The SOI is strongly anticorrelated with equatorial Pacific SSTs so that an El Niño warming event is associated with negative SOI values.

The NAO is characterized by fluctuations in SLP differences. Monthly values are an indicator of the strength and/or position of the subtropical Bermuda High. The relationship might result from a teleconnection between the mid-latitudes and tropics

whereby a below-normal NAO during the spring leads to dry conditions over the continents and to a tendency for greater summer–fall middle tropospheric ridging (enhancing the dry conditions). Ridging over the eastern and western sides of the North Atlantic basin tends to keep the middle tropospheric trough, responsible for hurricane recurvature, farther to the north during the peak of the season (Elsner and Jagger, 2006). The data sets containing the environmental variables are described in Chapter 6, where you also plotted them (see Figure 6.6). With the exception of the NAO index, the monthly values are averages over the 3 months of August through October. The NAO index is averaged over the 2 months of May and June.

7.3 BIVARIATE RELATIONSHIPS

Consider the relationship between hurricane frequency and one of the environmental variables (covariate). Scatter plots are not very useful as there are many covariate values for a given count. It is better to plot a summary of the covariate distribution for each count.

The five-number summary provides information about the median, the range, and the quartile values of a distribution (see Chapter 5). So, for example, you compare the five-number summary of the NAO during years with no hurricanes and during years with three hurricanes by typing

```
> load("annual.RData")
> nao0 = annual$nao[H$All == 0]
> nao3 = annual$nao[H$All == 3]
> fivenum(nao0); fivenum(nao3)
[1] -2.030 -0.765 -0.165  0.600  1.725
[1] -2.655 -1.315 -0.685 -0.085  1.210
```

For each quantile of the five-number summary, the NAO value is lower when there are three hurricanes compared to when there are no hurricanes. A plot showing the five-number summary values for all years and all covariates is shown in Figure 7.3. Note that the covariate is by convention plotted on the horizontal axis in a scatter plot, so you make the lines horizontal.

The plots show that the SOI and NAO are likely important in statistically explaining hurricane counts as there appears to be a systematic variation in counts across the range of values. The variation is less clear with SST and sunspot number. The bivariate relationships do not necessarily tell the entire story. A covariate might be important in explaining the residual variability so all the variables might be significant in a multivariate model (see Chapter 3).

7.4 POISSON REGRESSION

The model of choice for count data is Poisson regression. Poisson regression assumes that the response variable has a Poisson distribution, and the logarithm of the

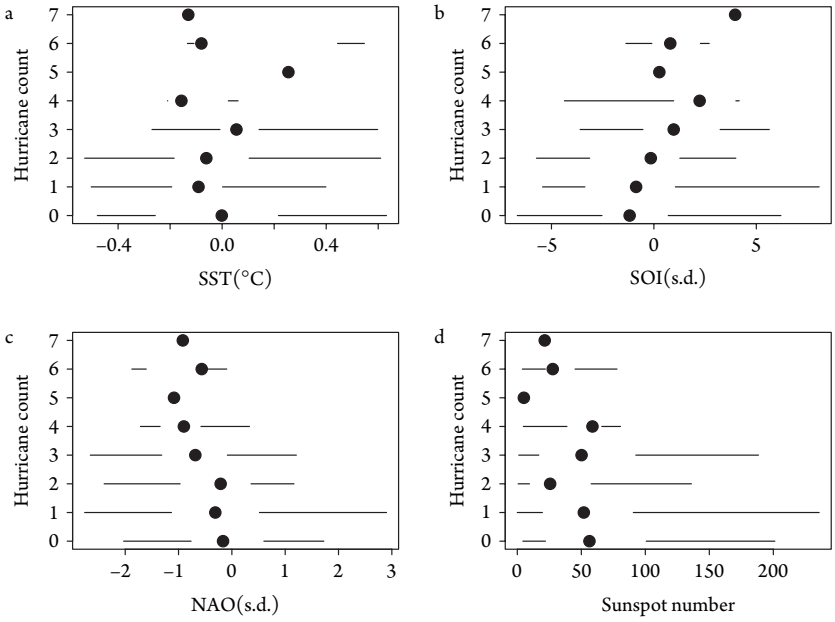

Figure 7.3 Bivariate relationships between covariates and hurricane counts.

expected value of the response variable is modeled with a linear combination of explanatory variables. It is an example of a log-linear model.

7.4.1 Limitation of Linear Regression

The linear regression model described in Chapter 3 is not appropriate for count data. To illustrate, here you regress U.S. hurricane counts on the four explanatory variables (covariates) described earlier. You then use the model to make predictions specifying the SOI and NAO at three standard deviation departures from the average, a large sunspot number, and an average SST value.

To make things a bit simpler, you first create a data frame by typing

```
> df = data.frame(All=H$All, SOI=annual$soi,
+   NAO=annual$nao, SST=annual$sst, SSN=annual$ssn)
> df = df[-(1:15), ]
```

Here the data frame object `df` has columns with labels `All`, `SOI`, `NAO`, `SST`, and `SSN` corresponding to the response variable U.S. hurricane counts and the four explanatory variables. You remove the first 15 years because of missing SOI values.

You then create a linear regression model object using the `lm` function specifying the response and covariates accordingly.

```
> lrm = lm(All ~ SOI + NAO + SST + SSN, data=df)
```

Your model is saved in `lrm`.

Next you use the `predict` method on the model object together with specific explanatory values specified using the `newdata` argument. The names must match those used in your model object, and each explanatory variable must have a value.

```
> predict(lrm, newdata=data.frame(SOI=-3, NAO=3,
+     SST=0, SSN=250))
     1
-0.318
```

The prediction results in a negative number that is not a count. It indicates that the climate conditions are unfavorable for hurricanes, but the number has no physical meaning.

7.4.2 Poisson Regression Equation

A Poisson regression model that specifies the logarithm of the annual hurricane rate is an alternative to linear regression. The assumption is that the hurricanes are independent in the sense that the arrival of one hurricane will not make another one more or less likely, but the rate of hurricanes varies from year to year due to the covariates.

The Poisson regression model is expressed as

$$\log(\lambda) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p \tag{7.2}$$

Here there are p covariates (indicated by the x_i's) and $p + 1$ parameters (β_i's). The model uses the logarithm of the rate as the response variable, but it is linear in the regression structure. It is not the same as a linear regression on the logarithm of counts. The model coefficients are determined by the *method of maximum likelihood*.

It is important to understand that with a Poisson regression you cannot explain all the variation in the observed counts; there will always be unexplainable variation due to the stochastic nature of the process. Thus even if the model precisely predicts the rate of hurricanes, the set of predicted counts will have a degree of variability that cannot be reduced by the model (aleatory uncertainty). Consequently, if you think that most of the variability in the counts can be explained, then Poisson regression is not the appropriate model.

7.4.3 Method of Maximum Likelihood

Given the set of parameters as a vector β and a vector of explanatory variables x, the mean of the predicted Poisson distribution is given by

$$E(Y|x) = e^{\beta' x} \tag{7.3}$$

and thus, the Poisson distribution's probability density function is given by

$$p(y|x;\beta) = \frac{e^{y(\beta' x)} e^{-e^{\beta' x}}}{y!} \tag{7.4}$$

Suppose that you are given a data set consisting of n vectors $x_i \in \mathbb{R}^{n+1}$, $i = 1, \ldots, n$, along with a set of n values $y_1, \ldots, y_n \in \mathbb{R}$. Then, for a given set of parameters β, the probability of attaining this particular set of data is given by

$$p(y_1, \ldots, y_n | x_1, \ldots, x_n; \beta) = \prod_{i=1}^{n} \frac{e^{y_i(\beta' x_i)} e^{-e^{\beta' x_i}}}{y_i!}. \tag{7.5}$$

By the method of maximum likelihood, you wish to find the set of parameters β that makes this probability as large as possible. To do this, the equation is first rewritten as a likelihood function in terms of β.

$$L(\beta | X, Y) = \prod_{i=1}^{n} \frac{e^{y_i(\beta' x_i)} e^{-e^{\beta' x_i}}}{y_i!}. \tag{7.6}$$

Note that the expression on the right-hand side of the equation has not changed. By taking logarithms, the equation is easier to work with. The log-likelihood equation is given by

$$\ell(\beta | X, Y) = \log L(\beta | X, Y) = \sum_{i=1}^{n} \left(y_i(\beta' x_i) - e^{\beta' x_i} - \log(y_i!) \right). \tag{7.7}$$

Notice that the β's only appear in the first two terms of the summation. Therefore, given that you are interested only in finding the best value for β, you can drop the $y_i!$ and write

$$\ell(\beta | X, Y) = \sum_{i=1}^{n} \left(y_i(\beta' x_i) - e^{\beta' x_i} \right). \tag{7.8}$$

This equation has no closed-form solution. However, the negative log-likelihood, $-\ell(\beta | X, Y)$, is a convex function, and so standard optimization or gradient ascent techniques can be applied to find the optimal value of β, for which the probability is maximum.

7.4.4 Model Fit

The method of maximum likelihood is employed in the `glm` function to determine the model coefficients. The Poisson regression model is a type of generalized linear model (GLM) in which the logarithm of the annual rate is a linear function of the covariates (predictors).

To fit a Poisson regression model to U.S. hurricanes and save the model as an object, type

```
> prm = glm(All ~ SOI + NAO + SST + SSN, data=df,
+     family="poisson")
```

The model formula is identical to what you used to fit the earlier linear regression model. The formula is read as "U.S. hurricane counts are modeled as a function of

Table 7.1 Coefficients of the Poisson regression model.

	Estimate	*Std. error*	*z value*	$Pr(>\|z\|)$
(Intercept)	0.5953	0.1033	5.76	0.0000
SOI	0.0619	0.0213	2.90	0.0037
NAO	−0.1666	0.0644	−2.59	0.0097
SST	0.2290	0.2553	0.90	0.3698
SSN	−0.0023	0.0014	−1.68	0.0928

SOI, NAO, SST, and SSN." Differences from the linear model fitting include the use of the `glm` function and the argument specifying `family="poisson"`.

You examine the model coefficients by typing

```
> summary(prm)
```

The model coefficients and the associated statistics are shown in Table 7.1.

As anticipated from the bivariate relationships, the SOI and SST variables are positively related to the rate of U.S. hurricanes, and the NAO and sunspot number are negatively related. You can see that the coefficient on SST is positive but not statistically significant. Both the SOI and NAO have coefficients that provide convincing evidence against the null hypothesis, while the coefficient on the SSN provides suggestive but inconclusive evidence against the null.

Statistical significance is based on a null hypothesis that the coefficient is zero (Chapter 3). The ratio of the estimated coefficient to its standard error (z-value) has an approximate standard normal distribution assuming the null is true. The probability of finding a z-value this extreme or more is your p-value. The smaller the p-value, the less support there is for the null hypothesis given your data and model.

You use the `plotmo` function from the **plotmo** package (Milborrow, 2011b) to plot your model's response when varying one covariate while holding the other covariates constant at their median values.

```
> require(plotmo)
> plotmo(prm)
```

The results are shown in Figure 7.4. As SOI and SST increase so does the hurricane rate. By contrast the rate decreases with increasing NAO and SSN. The curvature arises from the fact that the covariates are related to the counts through the logarithm of the rate. The confidence bands are based on pointwise ±2 standard errors. Note that the relatively large width for SOI values above 5 s.d. and for NAO values below −2 s.d.

7.4.5 Interpretation

Interpretation of the Poisson regression coefficients is different than the interpretation of the linear regression coefficients explained in Chapter 3. For example, the

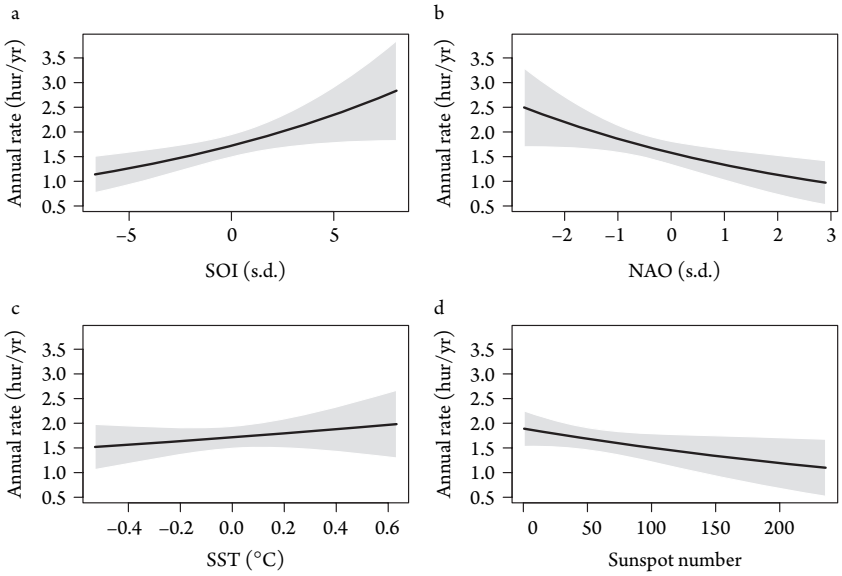

Figure 7.4 Dependence of hurricane rate on covariates in a Poisson regression.

coefficient value on the SOI indicates that for every one standard deviation (s.d.) increase in the SOI, the difference in the logarithm of hurricane rates is 0.062. Because there are other covariates, you must add "given that the other covariates in the model are held constant."

But what does the difference in the logarithm mean? Note that

$$\log A - \log B = \log\left(\frac{A}{B}\right) \tag{7.9}$$

so exponentiating the SOI coefficient value provides a ratio of the hurricane rates for a unit change in the SOI. You do this by typing

```
> exp(summary(prm)$coefficients[2, 1])
[1] 1.06
```

and find that for every one s.d. increase in SOI, the hurricane rate increases by a factor of 1.06 or 6 percent. Similarly, since the NAO coefficient value is -0.167, you find that for every one s.d. increase in the NAO, the hurricane rate decreases by a factor of 15 percent.

7.5 MODEL PREDICTIONS

Given the model coefficients obtained by the `glm` function and saved as a model object, you make predictions using the `predict` method. For comparison, you predict the rate of hurricanes given the same coefficient values used earlier for the linear regression model.

```
> predict(prm, newdata=data.frame(SOI=-3, NAO=3,
+   SST=0, SSN=250), type="response")
    1
0.513
```

The argument `type="response"` gives the prediction in terms of the mean response (hurricane rate). By default, `type="link"`, which results in a prediction in terms of the link function (here the logarithm of the mean response). Recall that the linear regression model gave a prediction that was physically unrealistic. Here, the predicted value indicates a small hurricane rate as you would expect given the covariate values, but the rate is a realistic nonnegative number.

The predicted rate together with Eq. 7.1 provides a probability for each possible count. To see this you create two bar plots, one for a forecast of hurricanes under unfavorable conditions and another for a forecast of hurricanes under favorable conditions. First you save the predicted rate for the specified values of the covariates. You then create a vector of counts from zero to six that is used as the set of quantiles for the `dpois` function and as the names argument in the `barplot` function. The plotting parameters are set using the `par` function. To make it easier to compare the probability distributions, limits on the vertical axis (`ylim`) are set the same.

```
> fav = predict(prm, newdata=data.frame(SOI=2, NAO=-2,
+   SST=0, SSN=50), type="response")
> ufa = predict(prm, newdata=data.frame(SOI=-2, NAO=2,
+   SST=0, SSN=200), type="response")
> h = 0:6
> par(mfrow=c(1, 2), las=1)
> barplot(dpois(x=h, lambda=ufa), ylim=c(0, .5),
+   names.arg=h, xlab="Number of Hurricanes",
+   ylab="Probability")
> mtext("a", side=3, line=1, adj=0, cex=1.1)
> barplot(dpois(x=h,lambda=fav), ylim=c(0,.5),
+   names.arg=h, xlab="Number of Hurricanes",
+   ylab="Probability")
> mtext("b", side=3, line=1, adj=0, cex=1.1)
```

The result is shown in Figure 7.5. The forecast probability of two or more hurricanes is 72 percent in years with favorable conditions but decreases to 16 percent in years with unfavorable conditions. The probability of no hurricanes during years with favorable conditions is 8 percent, which compares with a probability of 48 percent during years with unfavorable conditions. The model translates climate swings to changes in landfall probabilities.

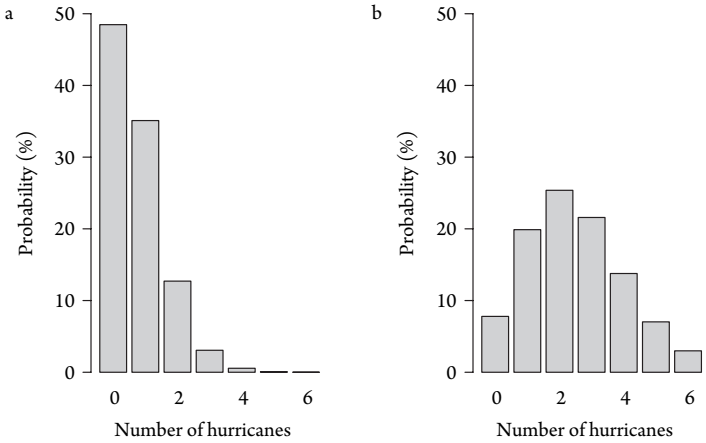

Figure 7.5 Forecast probabilities for (a) unfavorable and (b) favorable conditions.

7.6 FORECAST SKILL

7.6.1 Metrics

Forecast skill refers to how well predictions match observations. There are several ways to quantify this match. Here you consider three of the most common: the mean absolute error (MAE), the mean squared error (MSE), and the correlation coefficient (r).

Let λ_i be the predicted rate for year i and o_i be the corresponding observed count for that year. Then the three measures of skill over n years are defined by

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |\lambda_i - o_i| \tag{7.10}$$

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n} (\lambda_i - o_i)^2 \tag{7.11}$$

$$\text{r} = \frac{\sum(\lambda_i - \bar{\lambda})(o_i - \bar{o})}{\sqrt{\sum(\lambda_i - \bar{\lambda})^2 \sum(o_i - \bar{o})^2}} \tag{7.12}$$

You obtain the predicted rate for each year in the record (λ_i) by typing

```
> prm = glm(All ~ SOI + NAO + SSN, data=df,
+     family="poisson")
> pr = predict(prm, type="response")
```

You first create a new model removing the insignificant SST covariate. Since each prediction is made for a year with a known hurricane count, it is referred to as a "hindcast." The word "forecast" is reserved for a prediction made for a year where the hurricane count is unknown (typically in the future).

The Poisson regression hindcasts are given in terms of rates while the observations are counts. So instead of using a rate you use the probability distribution of observing

Table 7.2 Forecast skill (in sample). 'Useful' is the percentage skill above climatology.

	Poisson	*Climatology*	*Useful*
MAE	1.08	1.16	7.04
MSE	1.93	2.18	11.24
r	0.34		
MAEp	1.44	1.50	4.08
MSEp	3.67	3.92	6.26

$j = 0, 1, \ldots, \infty$ hurricanes. A probabilistic form of the earlier formulae are

$$\text{MAEp} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=0}^{\infty} \text{dpois}(j, \lambda_i) \cdot \left|j - o_i\right| \tag{7.13}$$

$$\text{MSEp} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=0}^{\infty} \text{dpois}(j, \lambda_i) \cdot (j - o_i)^2 \tag{7.14}$$

$$= \text{MSE} + \bar{\lambda} \tag{7.15}$$

where $\text{dpois}(j, \lambda_i)$ is the probability of j hurricanes in the ith year given the predicted rate λ_i, and o_i is the observed count. There is no probabilistic form to the correlation as a measure of forecast skill.

A prediction model is deemed useful if the skill level exceeds the level of a naive reference model. The percentage above greater than the skill obtained from a naive model is referred to useful skill. The naive model is typically climatology. To obtain the climatological rate, you type

```
> clim = glm(All ~ 1, data=df, family="poisson")
> cr = predict(clim, type="response")
```

Note that the only difference from your earlier is that the term to the right of the tilde is a 1. The model predicts the mean hurricane rate over the period of record. The value is the same for each year.

Table 7.2 shows skill metrics for your U.S. hurricane model and the percentage of useful skill relative to climatology. Correlation is undefined for a forecast of climatology. The useful skill level is between 4.1 and 11.2 percent. Although not high, it represents a significant improvement.

7.6.2 Cross-Validation

The earlier procedure results in an in-sample assessment of forecast skill. All years of data are used to estimate a single set of model coefficients with the model subsequently used to hindcast each year's hurricane activity. But how well will your model perform when making predictions of the future? This question is best

answered with an out-of-sample assessment of skill. An out-of-sample assessment (1) excludes a single year of observations, (2) determines the MLE coefficients of the Poisson regression model using observations from the remaining years, and (3) uses the model to predict the hurricane count for the year left out. This is done *n* times, removing each year's data successively. The above skill metrics are then used on these out-of-sample predictions. The procedure is called "crossvalidation," and where single cases are left out, it is called leave-one-out cross validation (LOOCV).

To perform LOOCV on your Poisson regression model, you loop over all years using the index `i`. Within the loop, you determine the model using all years except `i` (`df[-i, ]` in the `data` argument). You then make a prediction only for the year you left out (`newdata=df[i, ]`). Note that your climatology model is cross validated as well.

```
> j = 0:15; n = length(df$All)
> prx = numeric()
> crx = numeric()
> for(i in 1:n){
+ prm = glm(All ~ SOI + NAO + SSN,
+   data=df[-i, ], family="poisson")
+ clm = glm(All ~ 1, data=df[-i, ], family="poisson")
+ prx[i] = predict(prm, newdata=df[i, ], type="r")
+ crx[i] = predict(clm, newdata=df[i, ], type="r")
+ }
```

Skill assessment is done in the same way as for in-sample assessment. The results of the cross-validation assessment of model skill are give in Table 7.3. Out-of-sample skill levels are lower. This is an estimate of the average skill the model will have when making actual forecasts. You should show out-of-sample skill if you intend to use your model to predict the future.

The difference in percentage of usefulness between in-sample and out-of-sample skill is a measure of the overfit in your model. Overfit arises when your model

Table 7.3 Forecast skill (out of sample). 'Useful' in the percentage skill above climatology.

	Poisson	*Climatology*	*Useful*
MAE	1.11	1.16	4.91
MSE	2.05	2.21	7.03
r	0.26		
MAEp	1.46	1.51	2.81
MSEp	3.80	3.95	3.78

interprets random fluctuations as signal. Cross-validation helps you protect yourself against being fooled by this type of randomness.

7.7 NONLINEAR REGRESSION STRUCTURE

Poisson regression specifies a linear relationship between your covariates and the logarithm of the hurricane rate. This linearity can be restrictive if the influence of a covariate changes over the range of covariate valves. Multivariate adaptive regression splines (MARS) is a form of regression introduced by Friedman (1991) that allows for nonlinearity in the covariates.

MARS builds models of the form

$$\hat{f}(x) = \sum_{i=1}^{k} c_i B_i(x) \tag{7.16}$$

where $B_i(x)$ is a basis function and c_i is a constant for a given basis. The model is thus a weighted sum of the k basis functions. A basis function takes one of three forms: either a constant representing the intercept term, a *hinge* function of the form $\max(0, x - a)$, where a is a constant representing the *knot* for the hinge function, or a product of two or more hinge functions to allow the basis function the ability to handle interaction between two or more covariates. A hinge function is zero for part of its range, so it partitions your multivariate data into disjoint regions.

The `earth` function in the **earth** package (Milborrow, 2011a) provides functionality for MARS. The syntax is the same as other models in R. Here you create a model using MARS for your hurricane counts and environmental covariates by typing

```
> require(earth)
> mars = earth(All ~ SOI + NAO + SST + SSN, data=df,
+   glm=list(family="poisson"))
```

A summary method on the model object indicates that only the SOI and NAO are selected as important in explaining hurricane rates. The correlation between the predicted rate and the observed counts is obtained by taking the square root of the R-squared value.

```
> sqrt(mars$rsq)
[1] 0.469
```

This value exceeds the correlation from your Poisson regression by 39.7 percent, suggesting that MARS might be a better prediction model.

The partial dependence plot (Fig. 7.6) shows the hinge functions for the two selected covariates. The knots on the SOI are located at about −2 and +4 s.d. There is no relationship between the SOI and hurricane counts for the lowest values of SOI, and there is a sharp inverse relationship for the largest values. Caution is advised against overinterpretation as the graph describes only the SOI–hurricane relationship for median NAO values. The single knot on the NAO indicates no relationship between the NAO and hurricane counts for values less than about −0.5 s.d. Again,

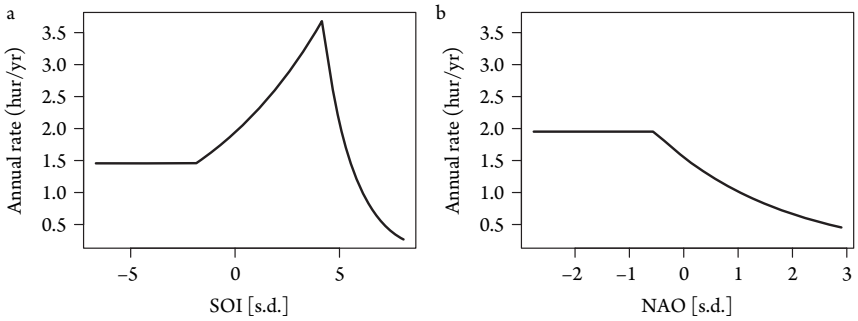

Figure 7.6 Dependence of hurricane rate on covariates in a MARS model.

this applies only at median SOI values. There are no standard errors from which to obtain confidence bands.

Before making forecasts, you use cross-validation on your MARS to get a correlation between observed and predicted using independent data. You do this by specifying the number of cross-validations with the `nfold` argument[1]. The function `earth` builds `nfold` cross-validated models. For each fold, it builds a model with in-sample data and then uses the model to compute the R-squared value from predictions made on the out-of-sample data. For instance, with `nfold=2`, the number of years in the in-sample and out-of-sample is roughly the same. The choice of years is chosen randomly, so you set a seed to allow replication of your results.

Here you set `nfold` at 5 as a compromise between having enough data to build the model and make predictions with it. You stabilize the variance further by specifying `ncross` to allow 40 different `nfold` cross-validations.

```
> set.seed(3042)
> marsCV = earth(All ~ SOI + NAO, data=df, nfold=5,
+   ncross=40, glm=list(family="poisson"))
```

The R-squared results are saved in your `marsCV` object in column `cv.rsq.tab`. The last row gives the mean R-squared value that provides an estimate of the average skill your model will have when it is used to make actual forecasts. The square root of that value is the correlation, obtained by typing

```
> rn = dim(marsCV$cv.rsq.tab)[1]
> mars.r = sqrt(marsCV$cv.rsq.tab[rn, ][1])
> mars.r
  All
0.291
```

The mean r value is 11 percent higher than the r value from the HOOCV of your Poisson regression model (see Table 7.3). This is still an improvement but below that estimated from your in-sample skill.

[1] Cross validation is done if the argument is greater than one.

7.8 ZERO-INFLATED COUNT MODEL

The Poisson regression model is a good place to start when working with count data, but it might not be good enough when the counts are overdispersed or when there are a large number of zeros. Consider the question of whether your Poisson regression of hurricane counts is adequate. You examine model adequacy using the residual deviance. The residual deviance is -2 times the log-likelihood ratio of a model without covariates compared to your model with covariates.

The residual deviance along with the residual degrees of freedom is available from the summary method on your `glm` object.

```
> prm = glm(All ~ SOI + NAO + SSN, data=df,
+   family="poisson")
> s = summary(prm)
> rd = s$deviance
> dof = s$df.residual
```

Under the null hypothesis that your model is adequate, the residual deviance has a χ^2 distribution with degrees of freedom equal to the residual degrees of freedom. Here the situation is reversed from the normal situation in which the null hypothesis is the opposite of what you hope for. To obtain the *p*-value for a test of model adequacy, type

```
> pchisq(rd, dof, lower.tail=FALSE)
[1] 0.0255
```

The residual deviance is 175.61 on 141 degrees of freedom resulting in a *p*-value of 0.0255. Thus there is evidence that something is missing.

The problem may be that hurricanes tend to arrive in clusters even after taking into account the covariates that influence hurricane rates from year to year. This clustering produces overdispersion in observed counts. You will examine this possibility and what to do about it in Chapter 11.

Another problem could be that the count data have too many zeros. This is typical when there are two processes at work: one determining whether there is at least one event and the other determining how many events. An example is the occurrence of cloud-to-ground lightning strikes. There will be many more hours with no strikes due to convective processes that are different than processes that produce one or more strikes. These kinds of data can be handled with zero-inflated models.

Zero-inflated count models are mixture models combining a point mass at zero and a count distribution. This leaves you with two sources of zeros: one from the point mass distribution and the other from the count distribution. Usually the count model is a Poisson regression and the point mass is a binomial regression.

The `zeroinfl` function in the **pscl** package (Zeileis et al., 2008) can be used to fit a zero-inflated model using the method of maximum likelihood. The formula describes the count data model, i.e., `y ~ x1 + x2` specifying a count data regression, where all zero counts have the same probability of belonging to the zero-inflation

component. This is equivalent to the model `y ~ x1 + x2 | 1`, making it explicit that the zero-inflated model has only an intercept. Additional predictors can be added so that not all zeros have the same probability of belonging to the point mass component or to the count component. A typical formula is `y ~ x1 + x2 | z1 + z2`. The covariates in the zero and the count component can be overlapping (or identical).

For example, to model your U.S. hurricane counts where the count model uses all four covariates and where the binomial model uses only the SST variable, type

```
> require(pscl)
> zim = zeroinfl(All ~ SOI + NAO + SST + SSN | SST,
+   data=df)
```

The model syntax includes a vertical bar to separate the covariates between the two model components. The returned object is of class `zeroinfl` and is similar to the object of class `glm`. The object contains a list of the coefficients and terms for each model component. To summarize the model object, type

```
> summary(zim)
Call:
zeroinfl(formula = All ~ SOI + NAO + SST + SSN |
    SST, data = df)

Pearson residuals:
   Min     1Q Median     3Q    Max
-1.492 -0.774 -0.127  0.583  3.061

Count model coefficients (poisson with log link):
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.64087    0.10593    6.05  1.5e-09
SOI          0.06920    0.02236    3.09   0.0020
NAO         -0.16928    0.06554   -2.58   0.0098
SST          0.57178    0.28547    2.00   0.0452
SSN         -0.00239    0.00143   -1.67   0.0942

Zero-inflation model coefficients (binomial with
   logit link):
            Estimate Std. Error z value Pr(>|z|)
(Intercept)    -4.44       2.03   -2.19    0.029
SST             6.80       3.89    1.75    0.080

Number of iterations in BFGS optimization: 16
Log-likelihood: -231 on 7 Df
```

Results show that the four covariates have a statistically significant influence on the number of U.S. hurricanes and that SST has a significant relationship with whether or

not there will be at least one hurricane. But note that the sign on the SST coefficient is positive in the zero-inflation component indicating that higher SST is associated with *more* years *without* hurricanes. The sign is also positive in the count component indicating that, given at least one hurricane, higher SST is associated with a higher probability of two or more hurricanes.

A cross-validation exercise indicates that this the zero-inflated model performs slightly worse than the Poisson model. The useful skill as measured by the mean absolute error is 3.7 percent above climatology for your zero-inflated model compared with 4.9 percent above climatology for your Poisson model.

7.9 MACHINE LEARNING

You can remove the Poisson assumption all together by employing a machine-learning algorithm that searches your data to find patterns related to the annual counts (data mining). A regression tree is a type of machine learning algorithm that outputs a series of decisions with each decision leading to a value of the response or to another decision. If the value of the NAO is less than −1 s.d., for example, then the response is two hurricanes. If it is greater, then is the SOI greater than 0.5 s.d. and so on. A single tree will capture the relationships between annual counts and your predictors.

To see how this works, import the annual hurricane data and subset on years since 1950. Create a data frame containing only the basin-wide hurricane counts and SOI and SST as the two predictors.

```
> load("annual.RData")
> dat = subset(annual, Year >= 1950)
> df = data.frame(H=dat$B.1, SOI=dat$soi, SST=dat$sst)
```

Then using the `tree` function from the **tree** package (Ripley, 2011), type

```
> require(tree)
> rt = tree(H ~ SOI + SST, data=df)
```

The model syntax is the same, with the response variable to the left of the tilde and the covariates to the right. To plot the regression tree, type

```
> plot(rt); text(rt)
```

Instead of interpreting the parameter values from a table of coefficients, you interpret a regression tree from an upside-down tree-like diagram. You start at the top. The first branch is a split on your SST variable at a value of 0.33. The split is a rule. Is the value of SST less than 0.33°C? If yes, branch to the left; if no, branch to the right. All splits work this way. Following on the right, the next split is on SOI. If SOI is greater than 0.12 s.d., then the mean value of all years under these conditions is 10.8 hur/yr. This is the end of the branch (leaf). You check this by typing

```
> mean(df$H[df$SST >= .33 & df$SOI > .12])
[1] 10.8
```

The model is fit using binary recursive partitioning. Splits are made along coordinate axes of SST and SOI so that on any branch, a split is chosen that maximally distinguishes the hurricane counts. Splitting continues until the variables cannot be split or there are too few years (less than 6 by default). Here, SST is the variable explaining the most variation in the counts so it gets selected first. Again, the value at which the split occurs is based on maximizing the difference in counts between the two subsets. The tree has five branches.

In general the key questions are as follows: which variables are best to use and which value gives the best split? The choice of variables is similar to the forward selection procedure of stepwise regression (Chapter 3). A prediction is made by determining which leaf is reached based on the values of your predictors. To determine the mean number of hurricanes when SOI is −2 s.d. and SST is 0.2°C, you use the `predict` method and type

```
> predict(rt, data.frame(SOI=-2, SST=.2))
   1
7.35
```

The predicted value depends on the tree and the tree depends on what years are used to grow it. For example, regrow the tree by leaving the last year out and make a prediction using the same two predictor values.

```
> rt2 = tree(H ~ SOI + SST, data=df[-61, ])
> predict(rt2, data.frame(SOI=-2, SST=.2))
   1
5.71
```

Results are different. Which prediction do you choose? Forecast sensitivity occurs with all statistical models, but it is more acute in models that contain a large number of parameters. Each branch in a regression tree is a parameter, so with your two predictors the model has five parameters.

A random forest algorithm sidesteps the question of prediction choice by making predictions from many trees (Breiman, 2001). It creates a sample from the set of all years and grows a tree using data only from the sampled years. It then repeats the sampling and grows another tree. Each tree gives a prediction and the mean is taken. The function `randomForest` in the **randomForest** package provides a random forest algorithm. For example, type

```
> require(randomForest)
> rf = randomForest(H ~ SOI + SST, data=df)
```

By default, the algorithm grows 500 trees. To make a prediction type,

```
> predict(rf, data.frame(SOI=-2, SST=.2))
   1
4.91
```

Regression trees and random forest algorithms tend to overfit your data, especially when you are searching over a large set of *potential* predictors in noisy climate data. Overfitting results in small in-sample error, but large out-of-sample error. Again, a cross-validation exercise is needed if you want to claim the algorithm has superior predictive skill. Cross-validation removes the noise specific to each year's set of observations and estimates how well the random forest algorithm finds prediction rules when this coincident information is unavailable. For example, *Does the random forest algorithm provide better prediction skill than a Poisson regression?* To answer this question, you arrange a HOOCV as follows:

```
> n = length(df$H)
> rfx = numeric(n)
> prx = numeric(n)
> for(i in 1:n){
+   rfm = randomForest(H ~ SOI + SST, data=df[-i, ])
+   prm = glm(H ~ SOI + SST, data=df[-i, ],
+             family="poisson")
+   new = df[i, ]
+   rfx[i] = predict(rfm, newdata=new)
+   prx[i] = predict(prm, newdata=new,
+                    type="response")
+ }
```

The out-of-sample mean-squared prediction error is computed by typing

```
> mean((df$H - prx)^2); mean((df$H - rfx)^2)
[1] 5.07
[1] 5.36
```

Results indicate that the Poisson regression performs slightly better than the random forest algorithm in this case although the difference is not statistically significant. The correlation between the actual and predicted value is 0.539 for the Poisson model and 0.502 for the random forest algorithm.

Figure 7.7 shows the bivariate influence of SST and SOI on hurricane counts using the random forest algorithm and Poisson regression. Hurricane counts increase with SST and SOI, but for high values of SOI, the influence of SST is stronger. Similarly for high values of SST, the influence of the SOI is more pronounced. The random forest is able to capture nonlinearities and thresholds but at the expense of interpreting some noise as signal, as seen by the relatively high count with SOI values near −3 s.d. and SST values near −0.1°C.

7.10 LOGISTIC REGRESSION

Some of our research in the 1990s focused on the climatology of hurricanes from nontropical origins (Elsner et al., 1996; Kimberlain and Elsner, 1998). We analyzed available information from each North Atlantic hurricane since 1944 to determine

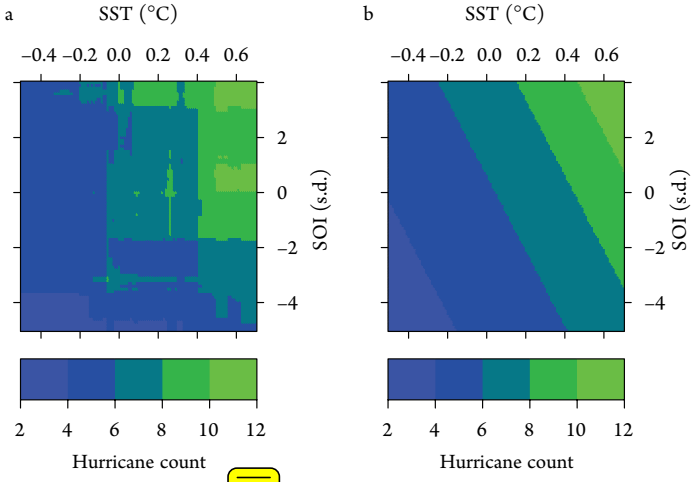

Figure 7.7 Hurricane resp[illegible] (a) Random forest and (b) Poisson regression.

whether we could discern middle-latitude influences on development. We classified hurricanes into tropical and baroclinic based on primary origin and development mechanisms. Here you would like to have a model that predicts a hurricane's group membership based simply on where the hurricane originated.

Logistic regression is the model of choice when your response variable is dichotomous. Many phenomena can be studied in this way. An event either occurs or it does not. The focus is to predict the occurrence of the event. A hurricane forecaster is keen about whether an area of disturbance will develop into a cyclone, given the present atmospheric conditions.

Logistic regression is a generalization of the linear regression model, where the response variable does not have a normal distribution and the regression structure is linear in the covariates. Like Poisson regression, the model coefficients are determined using the method of maximum likelihood.

The mean of a binary variable is a percentage. For example, generate 10 random binary values and compute the mean by typing

```
> set.seed(123)
> x = rbinom(n=10, size=1, prob=.5)
> x
 [1] 0 1 0 1 1 0 1 1 1 0
> mean(x)
[1] 0.6
```

Think of the 1s as heads and 0s as tails from 10 flips of a fair coin (`prob=.5`). You find 6 heads in 10 flips. The mean number is the percentage of heads in the sample. The percentage of a particular outcome can be interpreted as a probability so it is denoted as π. The logistic regression model specifies how π is related to a set of explanatory variables.

7.10.1 Exploratory Analysis

You input the hurricane data by typing

```
> bh = read.csv("bi.csv", header=TRUE)
> table(bh$Type)
  0   1   3
187  77  73
```

The type as determined in Elsner et al. (1996) is given by the variable `Type` with 0 indicating tropical-only, 1 indicating baroclinic influences, and 3 indicating baroclinic initiation. The typing was done subjectively using all the available synoptic information about each hurricane. While the majority of hurricanes form over warm ocean waters of the deep tropics ('tropical-only') some are aided in their formation by interactions with midlatitude jet stream winds ('baroclinically induced'). The stronger, tropical-only hurricanes develop farther south and primarily occur in August and September. The weaker, baroclinically induced hurricanes occur throughout the season.

First combine the baroclinic types into a single group and add this column to the data frame.

```
> bh$tb = as.integer(bh$Type != 0)
> table(bh$tb)
  0   1
187 150
```

With this grouping, there are 187 tropical and 150 baroclinic hurricanes in the record. Thus you can state that a hurricane drawn at random from this set of cyclones has about a 55 percent chance of being tropical-only.

Your interest is to improve on this climatological model by adding a covariate. Here you consider the latitude at which the cyclone first reaches hurricane strength. As an exploratory step, you create box plots of the latitudes grouped by hurricane type.

```
> boxplot(bh$FirstLat ~ bh$tb, horizontal=TRUE,
+   notch=TRUE, yaxt="n", boxwex=.4,
+    xlab="Latitude of Hurricane Formation")
> axis(2, at=c(1, 2), labels=c("Tropical",
+    "Baroclinic"))
```

Here you make the boxes horizontal (Fig. 7.8) because latitude is your explanatory variable. With the argument `notch` switched on, notches equal true, notches are drawn on the box sides. The vertical dash inside the box is the median latitude. Notches extend to $\pm 1.58 \times \text{IQR}/\sqrt{n}$, where IQR is the interquartile range (see Chapter 2) and n is the sample size. If the notches of two box plots do not overlap, this provides evidence that the two medians are statistically different (Chambers et al., 1983).

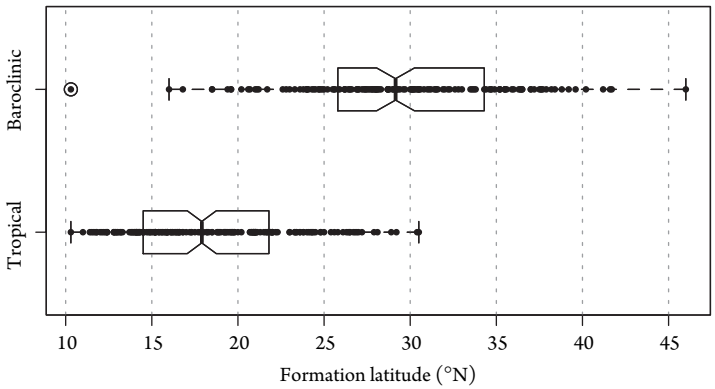

Figure 7.8 Genesis latitude by hurricane type.

The median formation latitude for the set of tropical hurricanes is 17.9°N, and for the set of baroclinic hurricanes, it is farther north at 29.1°N. This makes physical sense as cyclones farther south are less likely to have influence from middle-latitude baroclinic disturbances. The relatively small overlap between the two sets of latitudes strengthens your conviction that a latitude variable will improve a model for hurricane type.

7.10.2 Logit and Logistic Functions

Linear regression is not the appropriate model for binary data. It violates the assumption of equal variance and normality of residuals resulting in invalid standard errors and erroneous hypothesis tests. In its place, you use a generalized linear model as you did earlier with the count data.

However, instead of using the logarithm as the link between the response and the covariates as you did in the Poisson regression model, here you use the *logit* function. The logit of a number π between 0 and 1 is

$$\text{logit}(\pi) = \log\left(\frac{\pi}{1-\pi}\right) = \log(\pi) - \log(1-\pi) \tag{7.17}$$

If π is a probability then $\pi/(1-\pi)$ is the corresponding *odds*, and the logit of the probability is the logarithm of the odds. Odds are expressed as for:against (read: for to against) something happening. So the odds of a hurricane strike that is posted at 1:4 means there is a 20 percent change that a strike will occur.

The logistic regression model is expressed statistically as

$$\text{logit}(\pi) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p \tag{7.18}$$

where π is the mean. There are p covariates (x_i's) and $p+1$ parameters (β_i's).

To convert logit(π) to π (probability of occurrence), you use the *logistic* function (inverse of the logit function) given as

$$\text{logistic}(\alpha) = \frac{1}{1+\exp(-\alpha)} = \frac{\exp(\alpha)}{1+\exp(\alpha)} \tag{7.19}$$

7.10.3 Fit and Interpretation

To fit a logistic regression model to hurricane type with latitude as the covariate saving the model as an object, type

```
> lorm = glm(tb ~ FirstLat, data=bh,
+    family="binomial")
```

The call function is similar to Poisson regression, but here the family is binomial instead of poisson. The formula is read as "hurricane type is modeled as a function of formation latitude."

The model coefficients are determined by the method of maximum likelihood in the `glm` function. To produce a table of the coefficients, you type

```
> summary(lorm)$coefficients
             Estimate Std. Error z value Pr(>|z|)
(Intercept)    -9.083     0.9615   -9.45 3.50e-21
FirstLat        0.373     0.0395    9.45 3.49e-21
```

The estimated coefficients are listed by row. The coefficient for the intercept is the log odds of a baroclinic hurricane at the equator. In other words, the odds of being baroclinic when the latitude is zero is $\exp(-9.0826) = 0.000114$. These odds are very low, but that makes sense since no hurricanes form at the equator.

Interest is on the coefficient of the formation latitude variable indicated by the row labeled `FirstLat`. The value is 0.373. Before fitting the model, you anticipate the formation latitude coefficient to have a positive sign. Why? Because baroclinic (tropical)-type hurricanes are coded as 1 (0) in your data set and the box plots show that as formation latitude increases, the chance that a hurricane has baroclinic influences increases. Note that if your response values are character strings (e.g., "to" and "be") rather than coded as 0s and 1s, things will still work, but R will assign 0s and 1s based on alphabetical order and this will affect how you make sense of the coefficient's sign.

The magnitude of the coefficient is interpreted as for every degree increase in formation latitude, the log odds increases by a constant 0.373 units, on average. This is not very informative. By taking the exponent of the coefficient value, the interpretation is in terms of an odds ratio.

```
> exp(summary(lorm)$coefficients[2])
[1] 1.45
```

Thus, for every degree increase in formation latitude the odds ratio increases on average by a constant factor of 1.45 (or 45 percent). This 45 percent increase does not

depend on the value of latitude. That is, logistic regression is linear in the odds ratio. The interpretation is valid only over the range of latitudes in your data and physically meaningless for latitudes outside the range where hurricanes occur.

The table of coefficients includes the standard error and p-value. Statistical significance is based on a null hypothesis that the coefficient is zero. The ratio of the estimated coefficient to its standard error (z-value) has an approximate standard normal distribution assuming that the null is true. The probability of finding a z-value this extreme or more is the p-value. The smaller the p-value, the less support there is for the null hypothesis given the data and the model. The lack of support for the null allows you accept the model.

Also a confidence interval on the estimated coefficient is obtained by typing

```
> confint(lorm)[2, ]
 2.5 % 97.5 %
 0.301  0.456
```

This is interpreted to mean that although your best estimate for the log odds of a baroclinic hurricane given latitude is 0.373, there is a 95 percent chance that the interval between 0.301 and 0.456 will cover the true log odds.

7.10.4 Prediction

Predictions help you understand your model. As you did previously, you use the `predict` method on the model object. To predict the probability that a hurricane picked at random from your data will be baroclinic given that its formation latitude is 20°N latitude, you type

```
> predict(lorm, newdata=data.frame(FirstLat=20),
+    type="response")
    1
0.164
```

Thus, the probability of a baroclinic hurricane forming at this low latitude is 16.4 percent on average.

To create a plot of predictions across a range of latitudes, first prepare a vector of latitudes. The vector spans the latitudes in your data set. You specify an increment of 0.1° so the resulting prediction curve is smooth. You then use the predict method with `se.fit` equal to true and save the average prediction and the predictions corresponding to $\pm 1.96\times$ the standard error.

```
> lats = seq(min(bh$FirstLat), max(bh$FirstLat), .1)
> probs = predict(lorm,
+   newdata=data.frame(FirstLat=lats),
+   type="response", se.fit=TRUE)
> pm = probs$fit
```

```
> pu = probs$fit + probs$se.fit * 1.96
> pl = probs$fit - probs$se.fit * 1.96
```

Finally, you plot the data points at 0 and 1, as you did earlier, with the bar plot and add the predicted values using the `lines` function.

```
> plot(bh$FirstLat, bh$tb, pch=19, cex=.4,
+   ylab="Probability",
+   xlab="Formation Latitude (N)")
> grid()
> lines(lats, pm, lwd=2)
> lines(lats, pu, lwd=2, col="red")
> lines(lats, pl, lwd=2, col="red")
```

Results are shown in Figure 7.9. Tropical-only and baroclinically enhanced hurricane points are shown along the $y = 0$ and $y = 100$ lines, respectively. The gray band is the 95 percent pointwise confidence interval. Model predictions make sense. The probability of a baroclinically enhanced hurricane is less than 20 percent at latitudes south of 20°N. However, by latitude 24°N, the probability exceeds 50 percent and by latitude 31°N the probability exceeds 90 percent. Note that although the odds ratio is constant, the probability is a nonlinear function of latitude.

7.10.5 Fit and Adequacy

Output from a summary method applied to your model object (`summary(lorm)`) prints statistics of model fit including null and deviance residuals and the AIC (see Chapter 3). These are shown below the table of coefficients. One measure of model fit is the significance of the overall model.

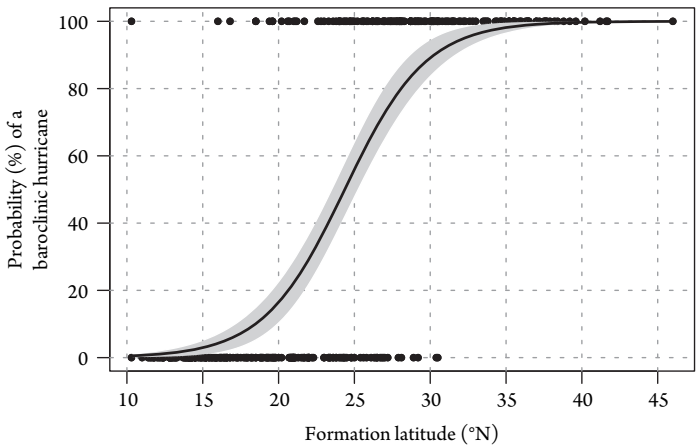

Figure 7.9 Logistic regression model for hurricane type.

This test asks whether the model with latitude fits significantly better than a model with an intercept only. An intercept-only model is called a "null" model (no covariates). The test statistic is the difference between the residual deviance for the model with and without latitude. The test statistic has a χ-squared distribution with degrees of freedom equal to the differences in degrees of freedom between the latitude model and the null model (i.e., the number of predictors in the model, here just one).

To find the difference in deviance between the two models (i.e., the test statistic) along with the difference in degrees of freedom, type

```
> dd = lorm$null.deviance - lorm$deviance
> ddof = lorm$df.null - lorm$df.residual
> dd; ddof
[1] 231
[1] 1
```

Then the *p*-value as evidence in support of the null model is obtained by typing

```
> 1 - pchisq(q=dd, df=ddof)
[1] 0
```

This leads you to reject the null hypothesis in favor of the model that includes latitude as a covariate.

A model can fit well but still be inadequate if it is missing an important predictor or if the relationship has a different form. Model adequacy is examined with the residual deviance statistic. The test is performed under the null hypothesis that the model is adequate (see §7.8). Under this hypothesis, the residual deviance has a χ-squared distribution with residual degrees of freedom. Thus, to test the model for adequacy, you type

```
> pchisq(q=lorm$deviance, df=lorm$df.residual)
[1] 4.24e-06
```

The small *p*-value indicates that the model is not adequate. So while formation latitude is a statistically significant predictor of baroclinic hurricanes, the model can be improved.

To try and improve things, you add another variable to the model. Here you create a new model adding the latitude at which maximum intensity first occurred (`MaxLat`) and examine the table of coefficients.

```
> lorm2 = glm(tb ~ FirstLat + MaxLat, data=bh,
+   family="binomial")
> summary(lorm2)$coefficients
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -8.560     0.9770   -8.76 1.93e-18
FirstLat       0.504     0.0662    7.62 2.50e-14
MaxLat        -0.134     0.0482   -2.77 5.57e-03
```

Although the latitude at maximum intensity is also statistically significant, something is wrong. The sign on the coefficient is negative indicating that baroclinic hurricanes are more likely if maximum latitude occurs farther south. This lacks physical sense, and it indicates a problem with the model.

The problem arises because of the high correlation between your two explanatory variables (see Chapter 3). You check the correlation between the two variables by typing

```
> cor(bh$FirstLat, bh$MaxLat)
[1] 0.855
```

The correlation exceeds 0.6, so it is best to remove one of your variables. You go back to your one-predictor model, but this time you use maximum latitude. You again check the model for statistical significance and adequacy and find both.

```
> lorm3 = glm(tb ~ MaxLat, data=bh, family="binomial")
> summary(lorm3)$coefficients
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -5.965     0.6760   -8.82 1.10e-18
MaxLat         0.207     0.0236    8.78 1.58e-18
> pchisq(q=lorm3$deviance, df=lorm3$df.residual)
[1] 0.543
```

Thus, you settle on a final model that includes the latitude at maximum intensity as the sole predictor.

7.10.6 Receiver Operating Characteristics

Your model predicts the probability that a hurricane has baroclinic influences given its latitude at lifetime maximum intensity. To make a decision from this forecast, you need to choose a threshold probability. For example, if the probability exceeds 0.5, then you predict a baroclinic hurricane.

Given your set of hindcast probabilities, one for each hurricane, and a threshold probability, you can create a two-by-two table of observed versus hindcast frequencies indicating how many times you correctly forecast baroclinic and tropical hurricanes. Here you do this using the `table` function on the logical vector of your predictions together with the vector of observed hurricane types.

```
> tab = table(predict(lorm3, type="response") > .5,
+   bh$tb)
> dimnames(tab) = list(Predicted=c("True",
+   "False"), Observed=c("BE", "TO"))
> tab
         Observed
Predicted  BE  TO
```

```
        True  147  51
        False  40  99
```

Note that you use the *dimnames* function on the table object to get the row and column names. The results show that of the 150 tropical-only hurricanes, 99 are predicted as such by the model using the threshold of 0.5 probability. Similarly, of the 187 baroclinic hurricanes, 147 are predicted as such by the model.

In this binary setup, 147 is the number of true positives, 40 is the number of false negatives, 51 is the number of false positives, and 99 is the number of true negatives. The *sensitivity* of your classification scheme is defined as the true-positive proportion given as 147/(147+40) = 79%. The *specificity* is defined as the true-negative proportion given as 99/(40+99) = 71 %.

Note that the values for sensitivity and specificity depend on your choice of threshold. For example, by increasing the threshold to 0.7, the sensitivity changes to 94 percent and the specificity to 90 percent.

The false-positive proportion is one minus the specificity. As you allow for more false positives, you can increase the sensitivity of your model. In the limit, if your model predicts all hurricanes to be BE, then it will be perfectly sensitive (it makes no mistakes in predicting every baroclinic hurricane as baroclinic), but it will not be specific enough to be useful.

A graph of the sensitivity versus the false-positive rate (1−specificity) as the decision threshold is varied is called an ROC curve. Here you use the code in **roc.R** (Peter DeWitt) to generate the ROC curves for your logistic model. Source the code by typing

```
> source("roc.R")
```

The `roc` function uses the model formula together with a set of training (testing) and validation data to generate ROC output. Here you use the `sample` function to take 168 random samples from the set of integers representing the sequence of hurricanes. The corresponding set of hurricanes is used for training and the remaining hurricanes are used for validation.

```
> set.seed(23456)
> idx = sample(dim(bh)[1], trunc(.5*dim(bh)[1]))
> out = roc(formula(lorm), data = bh[idx, ],
+   testing.data=bh[-idx, ], show.auc=FALSE)
```

The output is a list of three elements with the first two containing the area under the ROC curves for the test and validation data, respectively. To plot the curves, type

```
> out$plot
```

Figure 7.10 is made using `ggplot` (see Chapter 5) and shows the two ROC curves corresponding to the training and validation data sets. The area under the curve is an indication of the amount of model skill with the diagonal line (dotted) indicating no skill. To list the areas, type

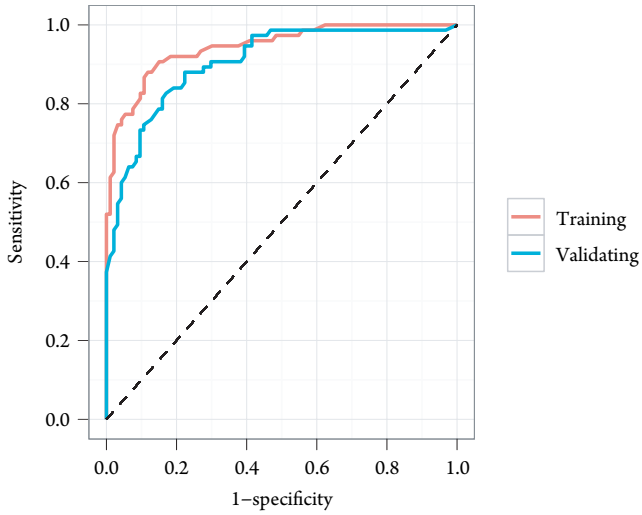

Figure 7.10 ROC curves for the logistic regression model of hurricane type.

```
> round(out$testing.auc, 3)
[1] 0.944
> round(out$validation.auc, 3)
[1] 0.905
```

Both values are close to 1 indicating a skillful model. In general, you expect the validation area to be less than the testing area.

You interpret the ROC curve as follows. Looking at the graph, if you allow a false-positive proportion of 20 percent (0.2 on the horizontal axis), then you can expect to correctly identify 84 percent of the future BE hurricanes. Since you are interested in future hurricanes, you use the validation curve. Note that if you want to perform better than that, say correctly identifying 95 percent of future BE hurricanes, then you need to accept a false-positive rate of about 40 percent.

This chapter showed how to build models for the occurrence of hurricanes. We began by modeling the annual counts of U.S. hurricanes with a Poisson regression and using environmental variables as covariates. We showed how to make predictions with the model and interpret the coefficients. We showed how to assess forecast skill including how to run a cross-validation exercise. We then showed how to include nonlinear terms in the regression with multivariate adaptive regression splines. We also took a look at a zero-inflated and a random forest model. We finished with an examination of logistic regression for predicting hurricane type. We showed how to interpret the coefficients, make predictions, and evaluate the ROCs when a decision threshold is imposed.