CHAPTER 8

Count Dependent Variables

Remember that we are going through all of these different types of regressions for two reasons: the assumptions of linear regression may not be met by certain types of dependent variables; and the type of dependent variable has information buried in its type.

This marks the third chapter of discrete dependent variables. In Chapter ??, we discussed binary dependent variables — dependent variables that can only take on two values. Last chapter, we discussed nominal and ordinal dependent variables. Nominal variables take on only a set number of values, which have no inherent relative value. Ordinal variables take on only a set number of values, which have an inherent value and ordering to them.

This chapter, we examine count dependent variables — dependent variables that can take on the value of any Natural number and whose value represents a count. Some examples of count variables include the number of fires in an area, the number of deaths due to terrorist attacks, and the number airplane flights taken by a person.

⋆ ⋆ ⋆

For count data, three things are important about that data: the variable can never be negative; the variable can have no theoretic upper bound; and the variable is discrete. Thus, if Y is a count

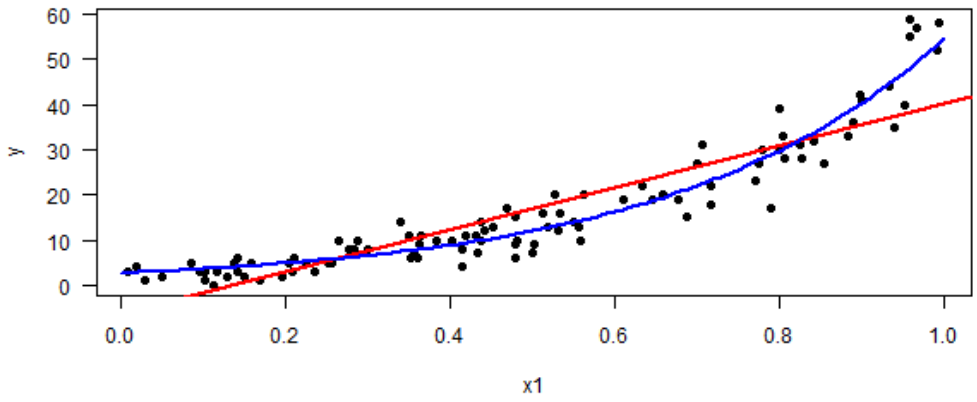

Figure 8.1: *Plot of the fabricated data with two regression equations overlaying. The linear regression is in red. The Poisson regression is in blue. Note that the blue line fits the data better than does the red line.*

variable, then $Y \in \mathbb{Z}^{\bullet}$, which can also be written as $Y \in \mathbb{N}$, the natural numbers. If we just do normal linear modeling without taking these three items into consideration, we lose information inherent in the data.[1] Performing count data analysis extracts more information from the data you worked so hard to collect.

8.1 Introductory example

Let us create a count dataset, fit it with a simple linear model, and then fit it with a Poisson model. The dataset that we will use for this example was fabricated so that we know the parameters. As such, we can compare the estimates we get from the two modeling techniques to the true parameters.

For this example, the true parameters are $\tilde{\beta}_0 = 1$ and $\tilde{\beta}_1 = 3$. Both of these are in log units. Thus, the 'real' parameters are $\beta_0 = 2.72$ and $\beta_1 = 20.09$. The estimates provided by the simple linear model are $\hat{\beta}_0 = -6.238$ and $\hat{\beta}_1 = 46.403$. The estimates provided by the Poisson regression are $\hat{\tilde{\beta}}_0 = 0.97$ and $\hat{\tilde{\beta}}_1 = 3.03$.

Beside the improvement on the estimation, there is an improvement in model fit. The Akaike

[1] We also stand a good chance of violating one of the assumptions of linear regression.

Information Criteria (AIC) score for the Poisson model is 523, whereas the AIC for the linear model is 651. Recall that a smaller AIC indicates a better fitting model (from Occam's perspective).

Next, we can examine the differences in the results by looking at the plots of the raw data and the two regression equations. The linear model (red line) clearly does not provide as good of a fit as the Poisson model (blue curve).

Finally, we can test the assumptions of our regression. According to the Shapiro-Wilk normality test, the residuals of the linear model are not Normally distributed ($W = 0.9494, p = 0.0007608$).

In all cases, the weakness of the linear model for this data are apparent. Not only are the parameter estimates off by large factors, but the model assumptions are not met. As such, if we know how the errors are distributed, we can use that to improve our knowledge about the underlying process.

The next section lays the mathematical groundwork for using a Poisson model to fit your data. It can be safely skimmed.

8.2 The mathematics

As the typical linear model is analyzed using a Gaussian distribution (a.k.a Normal distribution), because errors are distributed Gaussian, $\varepsilon \sim \mathcal{N}(\mu = 0, \sigma^2)$. The error terms in count data are not distributed as such. There are actually several possible named distributions for the error terms: Poisson, negative binomial, and quasi-Poisson are just a few.

As discussed in the Binary Dependent Variable chapter (Chapter **??**), the choice of the distribution depends on the error terms themselves. This is why we required the error terms to be Normally distributed when we were doing Ordinary Least Squares. Here, the typical count model is analyzed using a Poisson distribution. Equation 8.1 is the probability distribution function for the Poisson distribution. Note that it is a discrete distribution with possible values for x being any non-negative integer. As such, it is well-suited for dependent variables that are counts.

$$\mathcal{P}(x; \lambda) := \frac{e^{-\lambda}\lambda^x}{x!} \qquad \lambda > 0; x \in \{0, 1, 2, \ldots\} \tag{8.1}$$

An overlayed plot of several Poisson distributions with different parameter values is provided in Figure 8.2. Note the shape of a typical Poisson distribution. Smaller values for λ correspond to

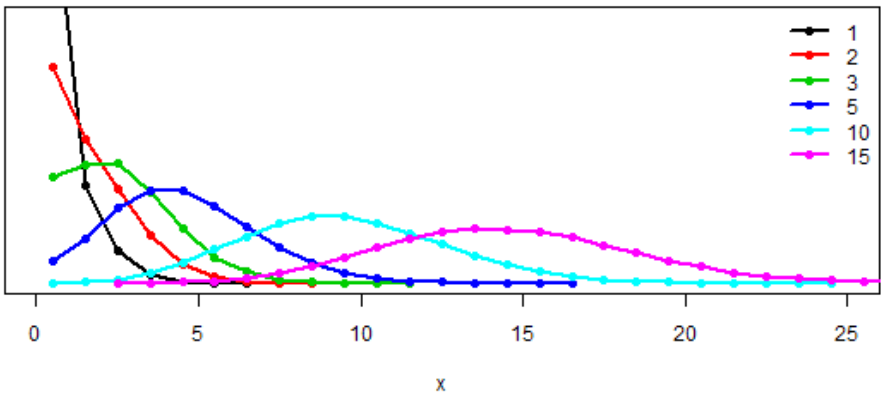

Figure 8.2: *Overlaid plots of several Poisson distributions. Note that as the parameter increases in value, the Poisson distribution approaches the Gaussian distribution.*

higher levels of right skewness. As such, when your expected counts are small, there may be large differences between the results of linear and Poisson regression.

We continue with our parameterization as we have in the past. In generalized linear models, we model the parameter, not the observed variable. The parameter here is λ. Thus, our linear predictor should be similar to $\lambda_i = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \cdots + \beta_k X_{i,k}$. Using matrix notation, this can be written as $\mathbf{\Lambda} = \mathbf{XB}$.

Note, however, that λ must be *non-negative*. Thus, we need a link function that turns a non-negative bounded variable into an unbounded variable. This link is the log link that we have seen in the past. Note that since we are modeling the parameter, and since the parameter is continuous, we can safely ignore the discreteness. Thus, our final equation modeling the parameter is $\log[\mathbf{\Lambda}] = \mathbf{XB}$, which is more easily seen as in Eqn 8.2:

$$\lambda_i = \exp\left[\beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \cdots + \beta_k X_{i,k}\right] \tag{8.2}$$

★ When doing generalized linear models, this all happens behind the scenes when you select the Poisson family and the log link. I just wanted you to understand why the log link was needed, and when the Poisson family is appropriate. The former is because the parameter must be non-negative.

The latter is to follow.

As the Poisson has just the one parameter, it is relatively inflexible. One particular feature of the Poisson is that its mean and variance are equal, and both are equal to its parameter, λ. Thus, a quick check to determine if the Poisson family is correct is to check the residual deviance and the residual degrees of freedom. If they are close, then the Poisson model is appropriate. If they are not close, then you should select a different family.

8.2.1 The quasi-Poisson family

The first alternative you may select is the quasi-Poisson family. It is identical to the Poisson family except that the variance is allowed to be a multiple of the mean.[2] This allows the quasi-Poisson to better estimate the standard errors in a Poisson model. You will note that the parameter estimates do not change; only the standard errors (and the statistical significance of those estimates) change.

The strength of the quasi-Poisson model is that its interpretation is identical to that of the Poisson model. The limitation is that few statistical programs are able to model using this family.

8.2.2 The negative binomial family

The second alternative you may select is the negative binomial family. The negative binomial family also allows for over- (and under-) dispersion in the model. It does this by assuming the parameter in the Poisson is distributed as a Gamma, with parameters α and β. The strength of this formulation is that a greater variety of variations are able to be fit. The drawback is that interpreting the results is a bit more difficult. However, since we make the computer do all the heavy lifting, this drawback is not too bad. It does, however, introduce a new set of possible error messages and parameters that you may have to interpret.

When the dispersion parameter for the Poisson is close to one, the results from the Poisson, the quasi-Poisson , and the negative binomial are extremely close in the parameter estimates and in the standard errors of those estimates.

In `R`, you will have to load the MASS library to use the negative binomial family, since it has its own regression function: `glm.nb`. In SPSS, `GENLIN` from the syntax file and "Analyze — Generalized Linear Models — Generalized Linear Models". In SAS, `GLM`. In STATA, `glm`.

[2] This is accomplished through a different fitting process that happens in the software.

8.3 Body counts

Using the above information, let us analyze a toy example with real data. This extended example will allow us to discuss a few things that are becoming important to our analyses.

Example 8.1. *The Trouble in Northern Ireland lasted from 1969 until 2002. During that time period, over 1800 people died as a result of terrorist actions. The prime ministers of the United Kingdom all had to deal with the several terrorist groups operating in Northern Ireland. If we assume that the terrorist groups are rational actors, then they will act to maximize their chances of achieving their goals. Because of its structure and size, the Provisional Irish Republican Army (PIRA) was best able to organize its actions to affect the elections.*

The research question is whether, and to what extent, the PIRA reacted to the political ideology of the current prime minister.

The literature is divided on the direction of the effect. Some research suggests that the PIRA became more violent and killed more people when the Conservatives held power. Other research suggests that the PIRA became more violent under the Labour party.

This current research examines whether the effect depends on the level of conservatism or liberalism in the prime minister.

The dataset contains just three variables of import: total (the total number of deaths under that prime minister for the year, or part of the year), days (the number of days during the year that the prime minister was in power), and riteleft (the level of conservatism of the prime minister). The second variable is necessary to control for the fact that some prime ministers only ruled for a part of the year. The third variable is the research variable. The first variable is the response variable (dependent variable). The basic research model is

```
deaths ~ riteleft
```

However, we need to deal with `days`, the number of days. It is here that we must make a decision. If we include `days` as a simple independent variable, we allow the effects of the `days` variable to freely vary. However, this may not really make sense. If the coefficient estimate for `days` is 2.35, what does that really mean?

It is probably better to treat `days` as the divisor for terrorist killings, thus ostensibly creating a

variable of `killings per day`. But, this is no longer a count model, nor is it a proportion model, as the average killings per day can be greater than 1. Fear not! Through the magic of mathematics, we can handle it.

Recall in Section 8.2 that the link function we used was the logarithm: $\log[\lambda] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$. If, instead of the expected count, λ, we wanted to model the expected ratio, $\frac{\lambda}{days}$, we have:

$$\log\left[\frac{\lambda}{\texttt{days}}\right] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

Using one of the properties of logarithms, this is equal to

$$\log[\lambda] - \log[\texttt{days}] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

This, in turn, is mathematically equivalent to

$$\log[\lambda] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \log[\texttt{days}]$$

As such, we now have a count model (the $\log[\lambda]$ is alone on the left) with an additional factor ($\log[days]$ on the right). Note that *there is no parameter* to estimate for $\log[days]$. This is important in how we set up the model, as `days` is not a typical variable. It is an *offset* variable.

Offset variables do not have parameters to estimate. They are direct effects with no multipliers. One can think of them as being subsumed in the constant term, which would be true if the offset variable did not vary. Most statistical programs have an offset option available when you specify the model to be fit. In `R`, the offset is specified in the model call by the keyword 'offset':

```
glm.nb(pira ~ riteleft, offset=log(days), data=terrorism)
```

Option 1: Independent variable

The first option is to treat the `days` variable as just another independent variable. This is not the best answer, as `days` has a specific meaning with respect to the number of terrorist deaths. The better option is to use Option 2 (below). However, for pedagogical purposes, we will enter `days` as an independent variable. Performing regressions for each of the three count data families, we get the summarized results in Table 8.1.

	Poisson		Quasi-Poisson		Negative Binomial	
Intercept	2.2622^{***}	0.1190	2.2622^{**}	0.7071	2.0363^{***}	0.6004
Conservatism	-0.0115^{***}	0.0011	-0.0115	0.0065	-0.0142	0.0093
Days in year	0.0050^{***}	0.0004	0.0050^{*}	0.0021	0.0058^{**}	0.0019
AIC	1482		NA		369	
Residual Deviance	1298		1298		46.9	

Table 8.1: *Results of three different families: Poisson, quasi-Poisson, and negative binomial. The numbers on the left of each column are the parameter estimates; on the right, the standard errors. The residual degrees of freedom are $\nu = 36$ for each model. The Akaike Information Criteria and the residual deviance are also provided for each model. Note the large amount of overdispersion in the Poisson model ($\frac{1298}{36} = 36$) indicating that the Poisson model is not an appropriate family for this model. Statistical significance: $*$: $0.05 \geq p > 0.01$; $**$: $0.01 \geq p > 0.001$; and $***$: $0.001 \geq p$.*

Note that the direction of each of the effects is the same. This is not always true, especially when the variable has little effect or has no statistical significance. However, if the variable is significant *and* changes effect direction, then there is something severely wrong with your research model. Also note that the effect is the same between the Poisson and the quasi-Poisson families. The only difference is the size of the standard errors. The quasi-Poisson will always give a better estimate of the standard errors (and the statistical significance) than the Poisson.

Note that the Poisson model is severely overdispersed — the residual deviance is much larger than the residual degrees of freedom. As such, the Poisson family would be (very) inappropriate for this model. Thus, either the quasi-Poisson or the negative binomial model would be preferable.

If we had just used the Poisson family, we would have concluded that the level of conservatism of the prime minister is highly significant. However, looking at the more-appropriate quasi-Poisson family, we see that the effect of conservatism is non-existent. Since the effect of conservatism on deaths was the purpose of this research question, it is very important to reach good conclusions.

As our research variable is not statistically significant at the usual level of significance, we will not even bother to predict and graph our predictions.

The `R` script that gives the information in this section is as follows:

```
model.1p <- glm(pira ~ riteleft + days, poisson, data=terrorism)
model.1q <- glm(pira ~ riteleft + days, quasipoisson, data=terrorism)
model.1n <- glm.nb(pira ~ riteleft + days, data=terrorism)

summary(model.1p)
summary(model.1q)
summary(model.1n)
```

	Poisson		Quasi-Poisson		Negative Binomial	
Intercept	-1.8280^{***}	0.0254	-1.8280^{**}	0.1495	3.8744^{***}	0.0969
Conservatism	-0.0106^{***}	0.0011	-0.0106	0.0063	-0.0069	0.0041
AIC	1480		NA		2080	
Residual Deviance	1297		1297		263.7	

Table 8.2: *Results of three different families: Poisson, quasi-Poisson, and negative binomial. The numbers on the left of each column are the parameter estimates; on the right, the standard errors. The residual degrees of freedom are $\nu = 37$ for each model. The Akaike Information Criteria and the residual deviance are also provided for each model. Note the large amount of overdispersion in the Poisson model ($\frac{1297}{37} = 35$) indicating that the Poisson family is not an appropriate family for this model. Statistical significance: $*: 0.05 \geq p > 0.01$; $**: 0.01 \geq p > 0.001$; and $***: 0.001 \geq p$.*

Option 2: Ratio

Option Two uses `days` as an exposure variable. This makes more sense than allowing it to freely enter the model as a typical independent variable. The results from fitting the data with the three model families are found in Table 8.2. While the final conclusions are the same as for Option 1, this need not always be true, especially as the parameter estimates for `days` in Option 1 (see Table 8.1) are not near 1.

According to the results in Table 8.2, the Poisson family is not appropriate; the level of overdispersion is high — on the order of 35. As such, a quasi-Poisson family would make a good substitute; the parameter estimates remain the same, but the estimates of the standard errors is changed to reflect the overdispersion. Thus, while the effect of conservatism was statistically significant in the Poisson model, it was not in the quasi-Poisson model.

The negative binomial model echoes the conclusions of the quasi-Poisson model. The level of conservatism has no discernable effect on the level of deaths resulting from PIRA terrorism in the United Kingdom during the Troubles in Northern Ireland.

The `R` script that gives the information in this section is as follows:

```
model.2p <- glm(pira ~ riteleft, offset=log(days), poisson, data=terrorism)
model.2q <- glm(pira ~ riteleft, offset=log(days), quasipoisson, data=terrorism)
model.2n <- glm.nb(pira ~ riteleft, offset(log(days)), data=terrorism)

summary(model.2p)
summary(model.2q)
summary(model.2n)
```

8.3.1 Next steps...

Using the results from both the quasi-Poisson and the negative binomial model does offer you the ability to strengthen your conclusions. If one result gave statistical significance and the other did not, then you would realize that your conclusions depended on the assumptions you made about the underlying mechanism that produced the data, and not on the variables you chose to include (or exclude). It is never a good place to find yourself when your substantive results depend on the choice between two acceptable models.

With that said, however, one should not stop here. Our formula is rather simplistic: it states that one independent variable is all we need to explain the dependent variable. It also assumes that the effect is linear between the independent and the dependent variable. If we believe that extremist prime ministers suffer from higher levels of terrorist killings, then the formula we have cannot capture that effect. To capture that effect, we will have to use the square (or higher powers) of the `riteleft` variable.

In fact, let us examine the effects of conservatism (up to the fourth power), plus the effects of having Labour in power, plus an interaction between having Labour in power and the level of conservatism in the Labour government. Thus, the research model we wish to fit will be Eqn 8.3:

$$\begin{aligned} \texttt{pira} = \beta_0 + \beta_1 \texttt{riteleft} + \beta_2 \texttt{riteleft}^2 + \beta_3 \texttt{riteleft}^3 + \beta_4 \texttt{riteleft}^4 \\ + \beta_5 \texttt{labour} + \beta_6 \texttt{labour} \times \texttt{riteleft} \end{aligned} \tag{8.3}$$

Of course, we would have to have good theory to provide this model to us, but let's just have fun with this.

In most statistical programs, one would have to create new variables for each of the powers (three new variables) and a new variable for the interaction term (`labour` × `riteleft`). In `R`, however, we can just write the formula to reflect what we want without having to worry about creating new variables. As such, in `R`, the formula will be

```
pira ~ riteleft + I(riteleft ^ 2) + I(riteleft ^ 3) + I(riteleft ^ 4)
    + labour + labour:riteleft
```

The use of `I` indicates that `R` should evaluate what is in the parentheses as a new variable. The

	Quasi-Poisson		Negative Binomial	
Intercept	-12.51^{**}	4.478	-6.980^{**}	2.396
Labour	-4.742^{**}	1.553	-4.843^{***}	0.2856
Conservatism	1.847^{*}	0.07101	1.866^{***}	0.3778
Conservatism^2	-0.03830^{*}	0.01425	-0.03833^{***}	0.0075
Conservatism^3	-0.002585^{**}	0.0009421	-0.002607^{***}	0.0005005
Conservatism^4	-0.00007314^{**}	0.00002642	-0.00007361^{***}	0.00001398
AIC	NA		1787	
Residual Deviance	334		230.5	

Table 8.3: *Results of two different families: quasi-Poisson and negative binomial. The numbers on the left of each column are the parameter estimates; on the right, the standard errors. The residual degrees of freedom are $\nu = 33$ for each model. The Akaike Information Criteria and the residual deviance are also provided for each model, when available. Statistical significance: $*: 0.05 \geq p > 0.01$; $**: 0.01 \geq p > 0.001$; and $***: 0.001 \geq p$.*

use of the colon indicates that we want to estimate the effect of the interaction between the two variables. Fitting this model using the quasi-Poisson family indicates that none of the terms have a statistically significant effect. This should not really surprise us, since there is a lot of correlation among the independent variables in that model. In the presence of high correlation, the standard errors tend to be larger than they should be.

Since nothing was statistically significant, let us pare the model to reduce the effect of correlation and get at some more basic effects. The best first thing to remove from the model is the interaction term. Doing this gives us the research model Eqn 8.4:

$$\begin{aligned} \texttt{pira} = \beta_0 &+ \beta_1 \texttt{riteleft} + \beta_2 \texttt{riteleft}^2 + \beta_3 \texttt{riteleft}^3 + \beta_4 \texttt{riteleft}^4 \\ &+ \beta_5 \texttt{labour} \end{aligned} \tag{8.4}$$

Fitting this model using both the quasi-Poisson family and the negative binomial family gives us the results in Table 8.3

Notice that all of our variables are now statistically significant. It turns out that the interaction term was so highly correlated with the other variables that it made it impossible to correctly estimate the effects of the research variables.

Now that we have two models that tell us, substantively, the same story, we should show the effect of the variables of interest. There are really only two independent variables involved here, with one being dichotomous. As such, we can show the effects on the same graph (one for each family). Figure 8.3 shows the predictions from both the quasi-Poisson model (Left Panel) and

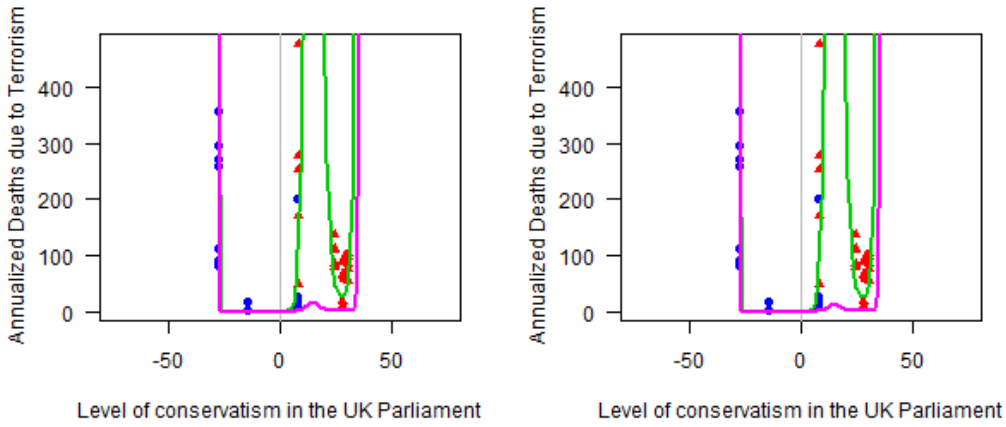

Figure 8.3: *Plot of the number of deaths due to terrorism, caused by the Provisional Irish Republican Army, in the United Kingdom during the Troubles in Northern Ireland. The points are overlayed with the quasi-Poisson model (Left Panel) and the negative binomial model (Right Panel). In both cases, the upper curve corresponds to the prediction when the Labour Party is not in power.*

the negative binomial model (Right Panel). The upper curve in both cases (green) corresponds to predictions when the Conservatives are in power.

The R script that gives the graphs in this section is as follows:

```
formula <- "pira ~ riteleft + I(riteleft^2) + I(riteleft^3) + I(riteleft^4) + labour"

model.3q <- glm(formula, offset=log(days), quasipoisson, data=terrorism)
model.3n <- glm.nb(formula, offset(log(days)), data=terrorism)
summary(model.3q)
summary(model.3n)

# Predictions
x <- -50:50
prediction.q0 <- exp(predict(model.3q, newdata=data.frame(riteleft=x, labour=0, days=365) ))
prediction.q1 <- exp(predict(model.3q, newdata=data.frame(riteleft=x, labour=1, days=365) ))
prediction.n0 <- exp(predict(model.3n, newdata=data.frame(riteleft=x, labour=0, days=365) ))
prediction.n1 <- exp(predict(model.3n, newdata=data.frame(riteleft=x, labour=1, days=365) ))

# Plot 1: quasi-Poisson
png("quasipoissonreg1.png", width=300, height=300)
plot(riteleft, total/days*365, type="n", xlim=c(-75,75), col=2, las=1,
        ylab="Annualized Deaths due to Terrorism",
        xlab="Level of conservatism in the UK Parliament" )
abline(v=0, col='grey')
points(riteleft[labour==1], total[labour==1]/days[labour==1]*365, col=4, pch=16)
points(riteleft[labour==0], total[labour==0]/days[labour==0]*365, col=2, pch=17)
```

```
lines(x,prediction.q0, col=3, lwd=2)
lines(x,prediction.q1, col=6, lwd=2)
dev.off()

# Plot 2: negative binomial
png("negbinreg1.png", width=300, height=300)
plot(riteleft, total/days*365, type="n", xlim=c(-75,75), col=2, las=1,
        ylab="Annualized Deaths due to Terrorism",
        xlab="Level of conservatism in the UK Parliament" )
abline(v=0, col='grey')
points(riteleft[labour==1], total[labour==1]/days[labour==1]*365, col=4, pch=16)
points(riteleft[labour==0], total[labour==0]/days[labour==0]*365, col=2, pch=17)
lines(x,prediction.n0, col=3, lwd=2)
lines(x,prediction.n1, col=6, lwd=2)
dev.off()
```

8.4 The Bias-Variance trade-off

Note that the predictions are *completely worthless*. Because we used so many parameters, the model fits the data (noise and all) as opposed to the underlying reality (signal). This is a common problem. Since our fit increases as we increase the number of variables in our models, there is a pressure for us to increase the number of variables. However, as in this case, using too many variables may tells us too little about the underlying process that gave rise to the data.

Remember that we are only using the data (a sample) to help us better understand the process that gave us the data (population). Fitting the data perfectly may actually tell us little about the process we are trying to model. However, not using enough variables may not get at the process, either. This trade-off between increasing the number of variables (which increases the reliance of the parameter estimates on the actual data) and reducing the number of variables (which increases the errors in our model) is termed the Bias-Variance trade-off, and it is a problem we must keep in our minds at all times. On the one hand, we want a good model that fits the population, on the other hand, we only know the sample (the data collected).

In the terrorism example above (Section 8.3), we can see that we used too many explanatory variables in our model. A glance at the graphs in Figure 8.3 suggests that we should have gone with a quadratic model (second power) at most, even though the quartic model (fourth power) fit the data better. Avoiding over-fitting the data is as simple as being aware of the dataset and the model predictions (of course, a good graph helps).

8.5 Conclusion

In this chapter, we examined what we can do when our dependent variable is a count variable. As counts are non-negative and discrete, nothing we have done thus far can properly handle them. While performing a log transform of the dependent variable as we did in Chapter **??** would allow us to actually make predictions that made sense, the resultant model would probably violate one or more of the assumptions of the classical linear model. Furthermore, merely transforming the data loses some of the information inherent in the fact that the data are counts.

Three model families were introduced to handle count data. The Poisson family requires that the mean and the variance be equal (which translates to the residual deviance and the residual degrees of freedom be equal). This is rarely the case. When the mean is much larger than the variance, the data are overdispersed. The other two families are used to handle overdispersed data.

All three families in this section model a parameter of the family and not the actual outcome. As the parameters must be non-negative, we use a log link to ensure this condition holds. Note that we are *not* transforming the dependent variable, we are transforming the family parameter — λ, in the case of the Poisson and the quasi-Poisson, and λ and θ for the negative binomial.

The last point of this chapter was to briefly discuss the Bias-Variance trade-off: including more variables fits the data better, but may not fit the underlying process that gave rise to the data, whereas fewer variables may miss both the data and the underlying process. There is a happy medium; we cannot know what it is.

⋆ ⋆ ⋆

This is the final chapter of the second part of this book. The next part deals with various topics that do not fit the regression paradigm as well as those in this part.

8.6 Extensions

1. Go back to the last model we fit (Eqn 8.4). Consider the comments about the model made in Section 8.4. Create a better model. Fit it with both the quasi-Poisson and the negative binomial. Plot graphs like those in Figure 8.3. Comment on the differences in the predictions between the two models.

8.7 R commands

offset The offset function (or function parameter) allows us to include known varying values in our regression. The variable included as an offset will not have an effect parameter estimated for it.

glm.nb() As negative binomial regression is fit using different methods, it cannot be included in the base `glm()` command. To access the `glm.nb()` command, you must include the MASS library in your script (`library(MASS)`). The output of the `glm.nb()` function is similar to that of the normal `glm()` command, with the inclusion of an estimate for θ and its standard error. If $\theta = 1$, then the Poisson model may be appropriate.