from it, we would stumble into the issue of assuming that the acceptance rate for one of the genders is pre-data more uncertain than the other. This isn't to say that over-parameterizing a model is always a good idea. But it isn't a violation of any statistical principle. You can always convert the posterior, post sampling, to any alternative parameterization. The only limitation is whether the algorithm we use to approximate the posterior can handle the high correlations. In this case, it can, and I bumped up the number iterations to make sure.

Rethinking: Simpson's paradox is not a paradox. This empirical example is a famous one in statistical teaching. It is often used to illustrate a phenomenon known as Simpson's paradox.[169] Like most paradoxes, there is no violation of logic, just of intuition. And since different people have different intuition, Simpson's paradox means different things to different people. The poor intuition being violated in this case is that a positive association in the entire population should also hold within each department. Overall, females in these data did have a harder time getting admitted to graduate school. But that arose because females applied to the hardest departments for anyone, male or female, to gain admission to.

Perhaps a little more paradoxical is that this phenomenon can repeat itself indefinitely within a sample. Any association between an outcome and a predictor can be nullified or reversed when another predictor is added to the model. And the reversal can reveal a true causal influence or rather just be a confound, as occurred in the grandparents example in Chapter 6. All that we can do about this is to remain skeptical of models and try to imagine ways they might be deceiving us. Thinking causally about these settings usually helps.[170]

11.2. Poisson regression

Binomial GLMs are appropriate when the outcome is a count from zero to some known upper bound. If you can analogize the data to the globe tossing model, then you should use a binomial GLM. But often the upper bound isn't know. Instead the counts never get close to any upper limit. For example, if we go fishing and return with 17 fish, what was the theoretical maximum? Whatever it is, it isn't in our data. How do we model the fish counts?

It turns out that the binomial model works here, provided we squint at it the right way. When a binomial distribution has a very small probability of an event p and a very large number of trials N, then it takes on a special shape. The expected value of a binomial distribution is just Np, and its variance is $Np(1-p)$. But when N is very large and p is very small, then these are approximately the same.

For example, suppose you own a monastery that is in the business, like many monasteries before the invention of the printing press, of copying manuscripts. You employ 1000 monks, and on any particular day about 1 of them finishes a manuscript. Since the monks are working independently of one another, and manuscripts vary in length, some days produce 3 or more manuscripts, and many days produce none. Since this is a binomial process, you can calculate the variance across days as $Np(1-p) = 1000(0.001)(1-0.001) \approx 1$. You can simulate this, for example over 10,000 (`1e5`) days:

R code
11.35

```
y <- rbinom(1e5,1000,1/1000)
c( mean(y) , var(y) )
```

```
[1] 0.9968400 0.9928199
```

The mean and the variance are nearly identical. This is a special shape of the binomial. This special shape is known as the Poisson distribution, and it is useful because it allows us to model binomial events for which the number of trials N is unknown or uncountably large. Suppose for example that you come to own, through imperial drama, another monastery. You don't know how many monks toil within it, but your advisors tell you that it produces, on average, 2 manuscripts per day. With this information alone, you can infer the entire distribution of numbers of manuscripts completed each day.

To build models with a Poisson distribution, the model form is even simpler than it is for a binomial or Gaussian model. This simplicity arises from the Poisson's having only one parameter that describes its shape, resulting in a data probability definition like this:

$$y_i \sim \text{Poisson}(\lambda)$$

The parameter λ is the expected value of the outcome y. It is also the expected variance of the counts y.

We also need a link function. The conventional link function for a Poisson model is the log link, as introduced in the previous chapter (page 326). So to embed a linear model, we use:

$$y_i \sim \text{Poisson}(\lambda_i)$$
$$\log(\lambda_i) = \alpha + \beta(x_i - \bar{x})$$

The log link ensures that λ_i is always positive, which is required of the expected value of a count outcome. But as mentioned in the previous chapter, it also implies an exponential relationship between predictors and the expected value. Exponential relationships grow very quickly, and few natural phenomena can remain exponential for long. So one thing to always check with a log link is whether it makes sense at all ranges of the predictor variables. The priors on the log scale also scale in surprising ways. So prior predictive simulation is again helpful.

11.2.1. Example: Oceanic tool complexity. The island societies of Oceania provide a natural experiment in technological evolution. Different historical island populations possessed tool kits of different size. These kits include fish hooks, axes, boats, hand plows, and many other types of tools. A number of theories predict that larger populations will both develop and sustain more complex tool kits. So the natural variation in population size induced by natural variation in island size in Oceania provides a natural experiment to test these ideas. It's also suggested that contact rates among populations effectively increase population size, as it's relevant to technological evolution. So variation in contact rates among Oceanic societies is also relevant.

We'll use this topic to develop a standard Poisson GLM analysis. And then I'll pivot at the end and also do a non-standard, but more theoretically motivated, Poisson model. The data we'll work with are counts of unique tool types for 10 historical Oceanic societies:[171]

R code 11.36

```
library(rethinking)
data(Kline)
d <- Kline
d
```

```
      culture population contact total_tools mean_TU
1    Malekula       1100     low          13     3.2
2     Tikopia       1500     low          22     4.7
3  Santa Cruz       3600     low          24     4.0
4         Yap       4791    high          43     5.0
```

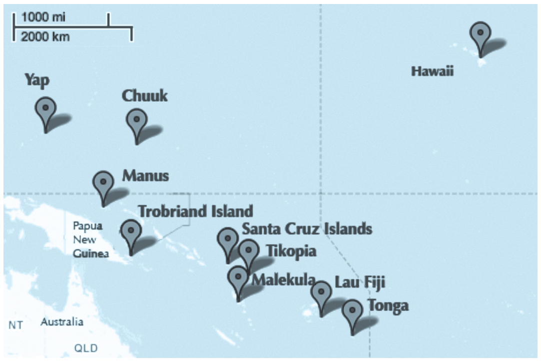

FIGURE 11.6. Locations of societies in the `Kline` data. The Equator and International Date Line are shown.

```
5    Lau Fiji       7400    high         33      5.0
6   Trobriand       8000    high         19      4.0
7       Chuuk       9200    high         40      3.8
8       Manus      13000     low         28      6.6
9       Tonga      17500    high         55      5.4
10     Hawaii     275000     low         71      6.6
```

That's the entire data set. You can see the location of these societies in the Pacific Ocean in FIGURE 11.6. Keep in mind that the number of rows is not clearly the same as the "sample size" in a count model. The relationship between parameters and "degrees of freedom" is not simple, outside of simple linear regressions. Still, there isn't a lot of data here, because there just aren't that many historic Oceanic societies for which reliable data can be gathered. We'll want to use regularization to damp down overfitting, as always. But as you'll see, a lot can still be learned from these data. Any rules you've been taught about minimum sample sizes for inference are just non-Bayesian superstitions. If you get the prior back, then the data aren't enough. It's that simple.

The `total_tools` variable will be the outcome variable. We'll model the idea that:

(1) The number of tools increases with the log `population` size. Why log? Because that's what the theory says, that it is the order of magnitude of the population that matters, not the absolute size of it. So we'll look for a positive association between `total_tools` and log `population`. You can get some intuition for why a linear impact of population size can't be right by thinking about mechanism. We'll think about mechanism more at the end.

(2) The number of tools increases with the `contact` rate among islands. No nation is an island, even when it is an island. Islands that are better networked may acquire or sustain more tool types.

(3) The impact of `population` on tool counts is moderated by high `contact`. This is to say that the association between `total_tools` and log `population` depends upon `contact`. So we will look for a positive interaction between log `population` and `contact` rate.

Let's build now. First, we make some new columns with the standardized log of `population` and an index variable for `contact`:

R code 11.37

```
d$P <- scale( log(d$population) )
d$contact_id <- ifelse( d$contact=="high" , 2 , 1 )
```

The model that conforms to the research hypothesis includes an interaction between log-population and contact rate. In math form, this is:

$$
\begin{aligned}
T_i &\sim \text{Poisson}(\lambda_i) \\
\log \lambda_i &= \alpha_{\text{CID}[i]} + \beta_{\text{CID}[i]} \log P_i \\
\alpha_j &\sim \text{to be determined} \\
\beta_j &\sim \text{to be determined}
\end{aligned}
$$

where P is `population` and CID is `contact_id`.

We need to figure out some sensible priors. As with binomial models, the transformation of scale between the scale of the linear model and the count scale of the outcome means that something flat on the linear model scale will not be flat on the outcome scale. Let's consider for example just a model with an intercept and a vague Normal(0,10) prior on it:

$$
\begin{aligned}
T_i &\sim \text{Poisson}(\lambda_i) \\
\log \lambda_i &= \alpha \\
\alpha &\sim \text{Normal}(0, 10)
\end{aligned}
$$

What does this prior look like on the outcome scale, λ? If α has a normal distribution, then λ has a log-normal distribution. So let's plot a log-normal with these values for the (normal) mean and standard deviation:

R code 11.38

```
curve( dlnorm( x , 0 , 10 ) , from=0 , to=100 , n=200 )
```

The distribution is shown in FIGURE 11.7 as the black curve. I've used a range from 0 to 100 on the horizontal axis, reflecting the notion that we know all historical tool kits in the Pacific were in this range. For the $\alpha \sim \text{Normal}(0, 10)$ prior, there is a huge spike right around zero—that means zero tools on average—and a very long tail. How long? Well the mean of a log-normal distribution is $\exp(\mu + \sigma^2/2)$, which evaluates to $\exp(50)$, which is impossibly large. If you doubt this, just simulate it:

R code 11.39

```
a <- rnorm(1e4,0,10)
lambda <- exp(a)
mean( lambda )
```

```
[1] 9.622994e+12
```

That's a lot of tools, enough to cover an entire island. We can do better than this.

I encourage you to play around with the `curve` code above, trying different means and standard deviations. The fact to appreciate is that a log link puts half of the real numbers—the negative numbers—between 0 and 1 on the outcome scale. So if your prior puts half its mass below zero, then half the mass will end up between 0 and 1 on the outcome scale. For Poisson models, flat priors make no sense and can wreck Prague. Here's my weakly informative suggestion:

R code 11.40

```
curve( dlnorm( x , 3 , 0.5 ) , from=0 , to=100 , n=200 )
```

I've displayed this distribution as well in FIGURE 11.7, as the blue curve. The mean is now $\exp(3 + 0.5^2/2) \approx 20$. We haven't looked at the mean of the `total_tools` column, and we

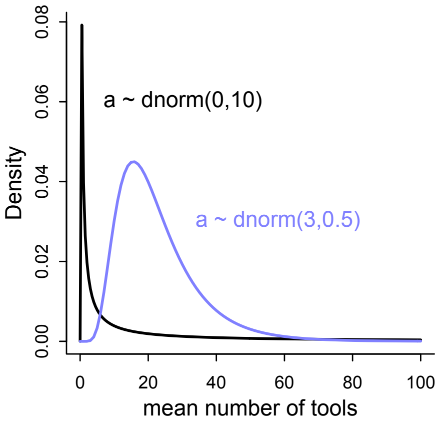

Figure 11.7. Prior predictive distribution of the mean λ of a simple Poisson GLM, considering only the intercept α. A flat conventional prior (black) creates absurd expectations on the outcome scale. The mean of this distribution is $\exp(50) \approx$ stupidly large. It is easy to do better by shifting prior mass above zero (blue).

don't want to. This is supposed to be a prior. We want the prior predictive distribution to live in the plausible outcome space, not fit the sample.

Now we need a prior for β, the coefficient of log population. Again for dramatic effect, let's consider first a conventional flat prior like $\beta \sim \text{Normal}(0, 10)$. Conventional priors are even flatter. We'll simulate together with the intercept and plot 100 prior trends of standardized log population against total tools:

R code 11.41

```
N <- 100
a <- rnorm( N , 3 , 0.5 )
b <- rnorm( N , 0 , 10 )
plot( NULL , xlim=c(-2,2) , ylim=c(0,100) )
for ( i in 1:N ) curve( exp( a[i] + b[i]*x ) , add=TRUE , col=grau() )
```

I display this prior predictive distribution as the top-left plot of Figure 11.8. The pivoting around zero makes sense—that's just the average log population. The values on the horizontal axis are z-score, because the variables is standardized. So you can see that this prior thinks that the vast majority of prior relationships between log population and total tools embody either explosive growth just above the mean log population size or rather catastrophic decline right before the mean. This prior is terrible. Of course you will be able to confirm, once we start fitting models, that even 10 observations can overcome these terrible priors. But please remember that we are practicing for when it does matter. And in any particular application, it could matter.

So let's try something much tighter. I'm tempted actually to force the prior for β to be positive. But I'll resist that temptation and let the data prove that to you. Instead let's just damping the prior's enthusiasm for impossibly explosive relationships. After some experimentation, I've settled on $\beta \sim \text{Normal}(0, 0.2)$:

R code 11.42

```
set.seed(10)
N <- 100
a <- rnorm( N , 3 , 0.5 )
b <- rnorm( N , 0 , 0.2 )
```

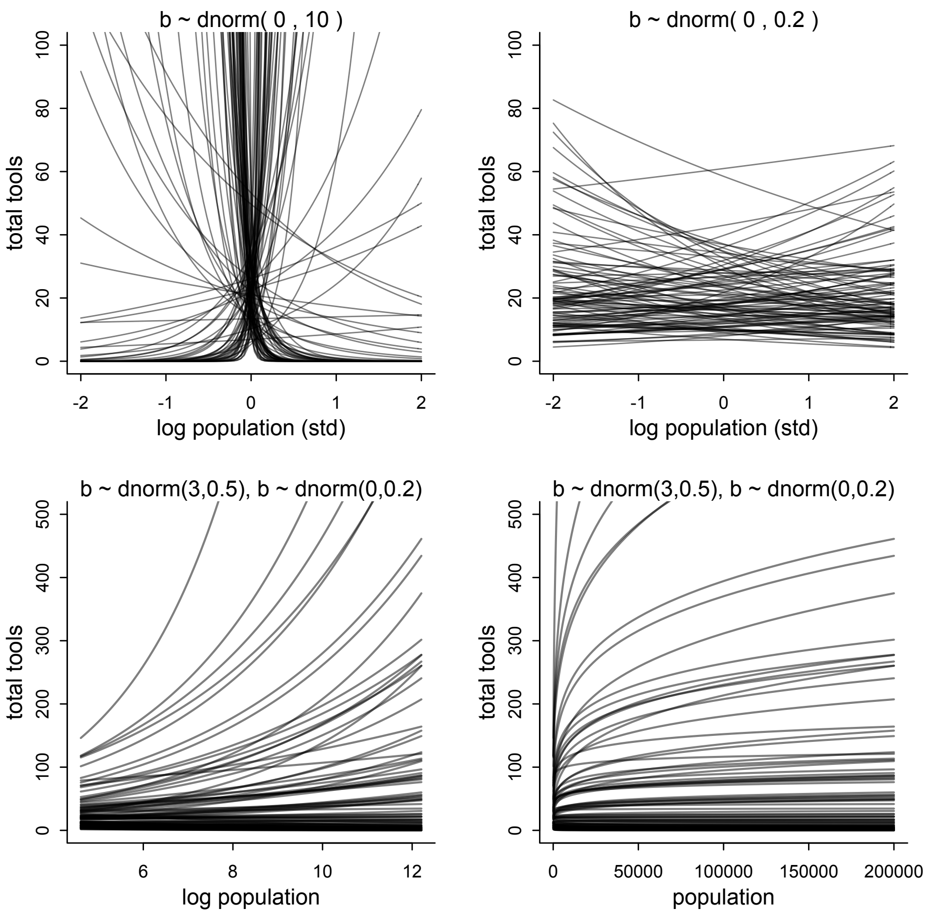

FIGURE 11.8. Struggling with slope priors in a Poisson GLM. Top-left: A flat prior produces explosive trends on the outcome scale. Top-right: A regularizing prior remains mostly within the space of outcomes. Bottom-left: Horizontal axis now on unstandardized scale. Bottom-right: Horizontal axis on natural scale (raw population size).

```
plot( NULL , xlim=c(-2,2) , ylim=c(0,100) )
for ( i in 1:N ) curve( exp( a[i] + b[i]*x ) , add=TRUE , col=grau() )
```

This plot is displayed in the top-right of FIGURE 11.8. Strong relationships are still possible, but most of the mass is for rather flat relationships between total tools and log population.

It will also help to view these priors on more natural outcome scales. The standardized log population variable is good for fitting. But it is bad for thinking. Population size has a natural zero, and we want to keep that in sight. Standardizing the variable destroys that. First, here are 100 prior predictive trends between total tools and un-standardized log population:

R code 11.43

```
x_seq <- seq( from=log(100) , to=log(200000) , length.out=100 )
lambda <- sapply( x_seq , function(x) exp( a + b*x ) )
plot( NULL , xlim=range(x_seq) , ylim=c(0,500) , xlab="log population" ,
    ylab="total tools" )
for ( i in 1:N ) lines( x_seq , lambda[i,] , col=grau() , lwd=1.5 )
```

This plot appears in the bottom-left of Figure 11.8. Notice that 100 total tools is probably the most we expect to ever see in these data. While most the of trends are in that range, some explosive options remain. And finally let's also view these same curves on the natural population scale:

R code 11.44

```
plot( NULL , xlim=range(exp(x_seq)) , ylim=c(0,500) , xlab="population" ,
    ylab="total tools" )
for ( i in 1:N ) lines( exp(x_seq) , lambda[i,] , col=grau() , lwd=1.5 )
```

This plot lies in the bottom-right of Figure 11.8. On the raw population scale, these curves bend the other direction. This is the natural consequence of putting the log of population inside the linear model. Poisson models with log links create **log-linear** relationships with their predictor variables. When a predictor variable is itself logged, this means we are assuming diminishing returns for the raw variable. You can see this by comparing the two plots in the bottom of Figure 11.8. The curves on the left would be linear if you log them. On the natural population scale, the model imposes diminishing returns on population: Each addition person contributes a smaller increase in the expected number of tools. The curves bend down and level off. Lots of predictor variables are better used as logarithms, for this reason. Simulating prior predictive distributions like these is a useful way to think through these issues.

Okay, finally we can approximate some posterior distributions. I'm going to code both the interaction model presented above as well as a very simple intercept-only model. The intercept only model is here because I want to show you something interesting about Poisson models and how parameters relate to model complexity. Here's the code for both models:

R code 11.45

```
dat <- list(
    T = d$total_tools ,
    P = d$P ,
    cid = d$contact_id )

# intercept only
m11.9 <- ulam(
    alist(
        T ~ dpois( lambda ),
        log(lambda) <- a,
        a ~ dnorm(3,0.5)
    ), data=dat , chains=4 , log_lik=TRUE )

# interaction model
m11.10 <- ulam(
    alist(
        T ~ dpois( lambda ),
        log(lambda) <- a[cid] + b[cid]*P,
        a[cid] ~ dnorm( 3 , 0.5 ),
        b[cid] ~ dnorm( 0 , 0.2 )
    ), data=dat , chains=4 , log_lik=TRUE )
```

Let's look at the PSIS model comparison quickly, just to flag two important facts.

R code 11.46

```
compare( m11.9 , m11.10 , func=PSIS )
```

```
         LOO pLOO dLOO weight    SE   dSE
m11.10  85.5  7.1  0.0      1 13.22    NA
m11.9  141.1  8.0 55.5      0 33.33 32.78
Warning messages:
1: Some Pareto k diagnostic values are too high.
```

First, note that we get the Pareto *k* warning again. This indicates some highly influential points. That shouldn't be surprising—this is a small dataset. But it means we'll want to take a look at the posterior predictions with that in mind. Second, while it's no surprise that the intercept-only model `m11.9` has a worse score than the interaction model `m11.10`, it might be very surprising that the "effective number of parameters" `pPSIS` is actually *larger* for the model with fewer parameters. Model `m11.9` has only one parameter. Model `m11.10` has four parameters. This isn't some weird thing about PSIS—WAIC tells you the same story. What is going on here?

The only place that model complexity—a model's tendency to overfit—and parameter count have a clear relationship is in a simple linear regression with flat priors. Once a distribution is bounded, for example, then parameter values near the boundary produce less overfitting than those far from the boundary. The same principle applies to data distributions. Any count near zero is harder to overfit. So overfitting risk depends both upon structural details of the model and the composition of the sample.

In this sample, a major source of overfitting risk is the highly influential point flagged by PSIS. Let's plot the posterior predictions now, and I'll scale and label the highly influential points with their Pareto *k* values. Here's the code to plot the data and superimpose posterior predictions for the expected number of tools at each population size and contact rate:

R code 11.47

```
k <- PSIS( m11.10 , pointwise=TRUE )$k
plot( dat$P , dat$T , xlab="log population (std)" , ylab="total tools" ,
    col=rangi2 , pch=ifelse( dat$cid==1 , 1 , 16 ) , lwd=2 ,
    ylim=c(0,75) , cex=1+normalize(k) )

# set up the horizontal axis values to compute predictions at
ns <- 100
P_seq <- seq( from=-1.4 , to=3 , length.out=ns )

# predictions for cid=1 (low contact)
lambda <- link( m11.10 , data=data.frame( P=P_seq , cid=1 ) )
lmu <- apply( lambda , 2 , mean )
lci <- apply( lambda , 2 , PI )
```

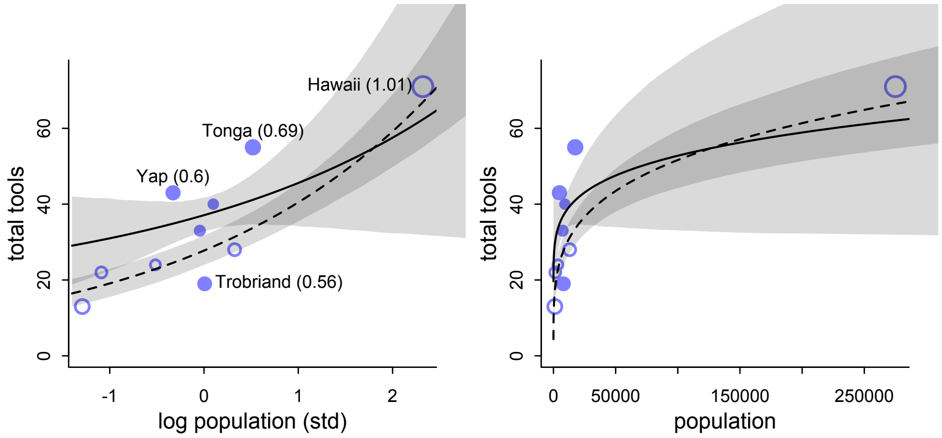

FIGURE 11.9. Posterior predictions for the Oceanic tools model. Filled points are societies with historically high contact. Open points are those with low contact. Point size is scaled by relative LOOIS Pareto *k* values. Larger points are more influential. The solid curve is the posterior mean for high contact societies. The dashed curve is the same for low contact societies. 89% compatibility intervals are shown by the shaded regions. Left: Standardized log population scale, as in the model code. Right: Same predictions on the natural population scale.

```
lines( P_seq , lmu , lty=2 , lwd=1.5 )
shade( lci , P_seq , xpd=TRUE )

# predictions for cid=2 (high contact)
lambda <- link( m11.10 , data=data.frame( P=P_seq , cid=2 ) )
lmu <- apply( lambda , 2 , mean )
lci <- apply( lambda , 2 , PI )
lines( P_seq , lmu , lty=1 , lwd=1.5 )
shade( lci , P_seq , xpd=TRUE )
```

The result is shown in FIGURE 11.9. Open points are low contact societies. Filled points are high contact societies. The points are scaled by their Pareto *k* values. The dashed curve is the low contact posterior mean. The solid curve is the high contact posterior mean.

This plot is joined on its right by the same predictions shown on the natural scale, with raw population sizes on the horizontal. The code to do that is very similar, but you need to convert the P_seq to the natural scale, by reversing the standardization, and then you can just replace P_seq with the converted sequence in the lines and shade commands.

R code 11.48

```
plot( d$population , d$total_tools , xlab="population" , ylab="total tools" ,
    col=rangi2 , pch=ifelse( dat$cid==1 , 1 , 16 ) , lwd=2 ,
    ylim=c(0,75) , cex=1+normalize(k) )

ns <- 100
```

```
P_seq <- seq( from=-5 , to=3 , length.out=ns )
# 1.53 is sd of log(population)
# 9 is mean of log(population)
pop_seq <- exp( P_seq*1.53 + 9 )

lambda <- link( m11.10 , data=data.frame( P=P_seq , cid=1 ) )
lmu <- apply( lambda , 2 , mean )
lci <- apply( lambda , 2 , PI )
lines( pop_seq , lmu , lty=2 , lwd=1.5 )
shade( lci , pop_seq , xpd=TRUE )

lambda <- link( m11.10 , data=data.frame( P=P_seq , cid=2 ) )
lmu <- apply( lambda , 2 , mean )
lci <- apply( lambda , 2 , PI )
lines( pop_seq , lmu , lty=1 , lwd=1.5 )
shade( lci , pop_seq , xpd=TRUE )
```

Hawaii ($k = 1.01$), Tonga ($k = 0.69$), Tap ($k = 0.6$), and the Trobriand Islands ($k = 0.56$) are highly influential points. Most are not too influential, but Hawaii is very influential. You can see why in the figure: It has extreme population size and the most tools. This is most obvious on the natural scale. This doesn't mean Hawaii is some "outlier" that should be dropped from the data. But it does mean that strongly Hawaii influences the posterior distribution. In the problems at the end of the chapter, I'll ask you do drop Hawaii and see what changes. For now, let's do something much more interesting.

Look at the posterior predictions in Figure 11.9. Notice that the trend for societies with high contact (solid) is higher than the trend for societies with low contact (dashed) with population size is low, but then the model allows it to actually be smaller. The means cross one another at high population sizes. Of course the model is actually saying it has no idea where the trend for high contact societies goes at high population sizes, because there are no high population size societies with high contact. There is only low-contact Hawaii. But it is still a silly pattern that we know shouldn't happen. A counter-factual Hawaii with the same population size but high contact should theoretically have at least as many tools as the real Hawaii. It shouldn't have fewer.

The model can produce this silly pattern, because it lets the intercept be a free parameter. Why is this bad? Because it means there is no guarantee that the trend for λ will pass through the origin where total tools equals zero and the population size equals zero. When there are zero people, there are also zero tools! As population increases, tools increase. So we get the intercept for free, if we stop and think.

Let's stop and think. Instead of the conventional GLM above, we could use the predictions of an actual model of the relationship between population size and tool kit complexity. By "actual model," I mean a model constructed specifically from scientific knowledge and hypothetical causal effects. The downside of this is that it will feel less like statistics—suddenly domain-specific skills are relevant. The upside is that it will feel more like science.

What we want is a dynamic model of the cultural evolution of tools. Tools aren't created all at once. Instead they develop over time. Innovation processes at them to a population. Processes of loss remove them. These forces balance to produce tool kits of different sizes.

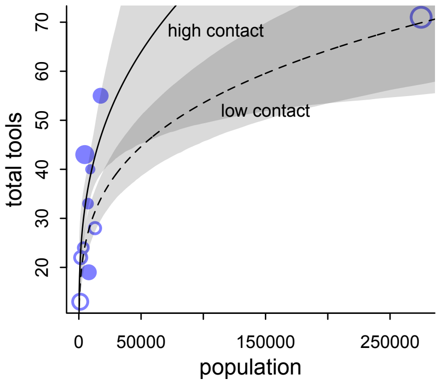

FIGURE 11.10. Posterior predictions for the scientific model of the Oceanic tool counts. Compare to the right hand plot in FIGURE 11.9. Since this model forces the trends to pass through the origin, as it must, its behavior is more sensible, in addition to having parameters with meaning outside a linear model.

The simplest model assumes that innovation is proportional to population size with some diminishing returns (an elasticity). It also assumes that tool loss is proportional to the number of tools, with no diminishing returns. If you recall the neutral evolution debate from Chapter 1, this is a model of that type—the theoretical model is more elaborate than the statistical model we'll derive from it.

The Overthinking box below presents the mathematical version of this model and shows you the code to build it in `ulam`. The model ends up in `m11.11`. Let's call this the *scientific model* and the previous `m11.10` the *geocentric model*. FIGURE 11.10 shows the posterior predictions for the scientific model, on the natural scale of population size. Comparing it with the analogous plot in FIGURE 11.9, notice that the trend for high contact societies always trends above the trend for low contact societies. Both trends always pass through the origin now, as they must. The scientific model is still far from perfect. But it provides a better foundation to learn from. The parameters have clearer meanings now. They aren't just bits of machinery in the bottom of a tide prediction engine.

You might ask how the scientific model compares to the geocentric model. The expected accuracy out of sample, whether you use PSIS or WAIC, is a few points better than the geocentric model. It is still tugged around by Hawaii and Tonga. We'll return to these data in a later chapter and approach contact rate a different way, by taking account of how close these societies are to one another.

Overthinking: Modeling tool innovation. Taking the verbal model in the main text above, we can write that the change in the expected number of tools in one time step is:

$$\Delta T = \alpha P^{\beta} - \gamma T$$

where P is the population size, T is the number of tools, and α, β, and γ are parameters to be estimated. To find an equilibrium number of tools T, just set $\Delta T = 0$ and solve for T. This yields:

$$\hat{T} = \frac{\alpha P^{\beta}}{\gamma}$$

We're going to use this inside a Poisson model now. The noise around the outcome will still be Poisson, because that is still the maximum entropy distribution in this context—`total_tools` is a count with no clear upper bound. But the linear model is gone:

$$T_i \sim \text{Poisson}(\lambda_i)$$
$$\lambda_i = \alpha P_i^{\beta}/\gamma$$

Notice that there is no link function! All we have to do to ensure that λ remains positive is to make sure the parameters are positive. In the code below, I'll use exponential priors for β and γ and a log-Normal for α. Then they all have to be positive. In building the model, we also want to allow some or all of the parameters to vary by contact rate. Since contact rate is suppose to mediate the influence of population size, let's allow α and β. It could also influence γ, because trade networks might prevent tools from vanishing over time. But we'll leave that as an exercise for the reader. Here's the code:

R code 11.49

```
dat2 <- list( T=d$total_tools, P=d$population, cid=d$contact_id )
m11.11 <- ulam(
    alist(
        T ~ dpois( lambda ),
        lambda <- exp(a[cid])*P^b[cid]/g,
        a[cid] ~ dnorm(1,1),
        b[cid] ~ dexp(1),
        g ~ dexp(1)
    ), data=dat2 , chains=4 , log_lik=TRUE )
```

I've invented the exact priors behind the scenes. Let's not get distracted with those. I encourage you to play around. The lesson here is in how we build in the predictor variables. Using prior simulations to design the priors is the same, although easier now that the parameters mean something. Finally, the code to produce posterior predictions is no different than the code in the main text used to plot predictions for `m11.10`.

11.2.2. Negative binomial (gamma-Poisson) models. Typically there is a lot of unexplained variation in Poisson models. Presumably this additional variation arises from unobserved influences that vary from case to case, generating variation in the true λ's. Ignoring this variation, or *rate heterogeneity*, can cause confounds just like it can for binomial models. So a very common extension of Poisson GLMs is to swap the Poisson distribution for something called the **negative binomial** distribution. The is really a Poisson distribution in disguise, and it is also sometimes called the **gamma-Poisson** distribution for this reason. It is a Poisson in disguise, because it is a mixture of different Poisson distributions. We'll work with mixtures in the next chapter.

11.2.3. Example: Exposure and the offset. The parameter λ is the expected value of a Poisson model, but it's also commonly thought of as a rate. Both interpretations are correct, and realizing this allows us to make Poisson models for which the **exposure** varies across cases *i*. Suppose for example that a neighboring monastery performs weekly totals of completed manuscripts while your monastery does daily totals. If you come into possession of both sets of records, how could you analyze both in the same model, given that the counts are aggregated over different amounts of time, different exposures?

Here's how. Implicitly, λ is equal to an expected number of events, μ, per unit time or distance, τ. This implies that $\lambda = \mu/\tau$, which lets us redefine the link:

$$y_i \sim \text{Poisson}(\lambda_i)$$
$$\log \lambda_i = \log \frac{\mu_i}{\tau_i} = \alpha + \beta x_i$$

Since the logarithm of a ratio is the same as a difference of logarithms, we can also write:

$$\log \lambda_i = \log \mu_i - \log \tau_i = \alpha + \beta x_i$$

These τ values are the "exposures." So if different observations i have different exposures, then this implies that the expected value on row i is given by:

$$\log \mu_i = \log \tau_i + \alpha + \beta x_i$$

When $\tau_i = 1$, then $\log \tau_i = 0$ and we're back where we started. But when the exposure varies across cases, then τ_i does the important work of correctly scaling the expected number of events for each case i. So you can model cases with different exposures just by writing a model like:

$$\begin{aligned} y_i &\sim \text{Poisson}(\mu_i) \\ \log \mu_i &= \log \tau_i + \alpha + \beta x_i \end{aligned}$$

where τ is a column in the data. So this is just like adding a predictor, the logarithm of the exposure, without adding a parameter for it. There will be an example later in this section. You can also put a parameter in front of $\log \tau_i$, which is one way to model the hypothesis that the rate is not constant with time.

For the last Poisson example, we'll look at a case where the exposure varies across observations. When the length of observation, area of sampling, or intensity of sampling varies, the counts we observe also naturally vary. Since a Poisson distribution assumes that the rate of events is constant in time (or space), it's easy to handle this. All we need to do, as explained above, is to add the logarithm of the exposure to the linear model. The term we add is typically called an *offset*.

We'll simulate for this example, both to provide another example of dummy-data simulation as well as to ensure we get the right answer from the offset approach. Suppose, as we did earlier, that you own a monastery. The data available to you about the rate at which manuscripts are completed is totaled up each day. Suppose the true rate is $\lambda = 1.5$ manuscripts per day. We can simulate a month of daily counts:

R code 11.50

```
num_days <- 30
y <- rpois( num_days , 1.5 )
```

So now y holds 30 days of simulated counts of completed manuscripts.

Also suppose that your monastery is turning a tidy profit, so you are considering purchasing another monastery. Before purchasing, you'd like to know how productive the new monastery might be. Unfortunately, the current owners don't keep daily records, so a head-to-head comparison of the daily totals isn't possible. Instead, the owners keep weekly totals. Suppose the daily rate at the new monastery is actually $\lambda = 0.5$ manuscripts per day. To simulate data on a weekly basis, we just multiply this average by 7, the exposure:

R code 11.51

```
num_weeks <- 4
y_new <- rpois( num_weeks , 0.5*7 )
```

And new y_new holds four weeks of counts of completed manuscripts.

To analyze both y, totaled up daily, and y_new, totaled up weekly, we just add the logarithm of the exposure to linear model. First, let's build a data frame to organize the counts and help you see the exposure for each case:

R code
11.52

```
y_all <- c( y , y_new )
exposure <- c( rep(1,30) , rep(7,4) )
monastery <- c( rep(0,30) , rep(1,4) )
d <- data.frame( y=y_all , days=exposure , monastery=monastery )
```

Take a look at `d` and confirm that there are three columns: The observed counts are in `y`, the number of days each count was totaled over are in `days`, and the new monastery is indicated by `monastery`.

To fit the model, and estimate the rate of manuscript production at each monastery, we just compute the log of each exposure and then include that variable in linear model. This code will do the job:

R code
11.53

```
# compute the offset
d$log_days <- log( d$days )

# fit the model
m11.12 <- quap(
    alist(
        y ~ dpois( lambda ),
        log(lambda) <- log_days + a + b*monastery,
        a ~ dnorm( 0 , 1 ),
        b ~ dnorm( 0 , 1 )
    ), data=d )
```

To compute the posterior distributions of λ in each monastery, we sample from the posterior and then just use the linear model, but without the offset now. We don't use the offset again, when computing predictions, because the parameters are already on the daily scale, for both monasteries.

R code
11.54

```
post <- extract.samples( m11.12 )
lambda_old <- exp( post$a )
lambda_new <- exp( post$a + post$b )
precis( data.frame( lambda_old , lambda_new ) )
```

```
'data.frame': 10000 obs. of 2 variables:
           mean   sd 5.5% 94.5%     histogram
lambda_old 1.34 0.21 1.03  1.70     ▁▁▃▇▅▂▁▁▁
lambda_new 0.52 0.14 0.33  0.77 ▁▁▃▇▇▃▂▁▁▁▁▁▁
```

The new monastery produces about half as many manuscripts per day. So you aren't going to pay that much for it.

11.3. Multinomial and categorical models

The binomial distribution is relevant when there are only two things that can happen, and we count those things. In general, more than two things can happen. For example, recall the bag of marbles from way back in Chapter 2. It contained only blue and white marbles. But suppose we introduce red marbles as well. Now each draw from the bag can be one of