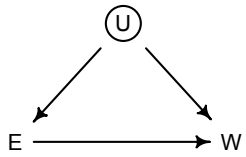
to an outcome. Some of those paths are causal, so we want to leave them open. Other paths are non-causal, for example back-door paths. We want to close those, as well as not accidentally open them by including the wrong variables in the model.

Of course sometimes it won't be possible to close all of the non-causal paths. What can be done in that case? More than nothing. If you are lucky, there are ways to exploit a combination of natural experiments and clever modeling that allow causal inference even when non-causal paths cannot be closed.

We'll start with the most famous, and possibly least intuitive, example. Then we'll move on to describe some other approaches.

14.3.1. Instrumental variables. Consider the impact of education E on wages W. Does more school improve future wages? If we just regress wages on achieved education, we expect the inference to be biased by factors that influence both wages and education. For example, industrious people may both complete more education and earn higher wages, generating a correlation between education and wages. But that doesn't necessarily mean that education causes higher wages. It is often difficult to measure, or even imagine, all of the possible confounds of this kind. We end up with a DAG like this:



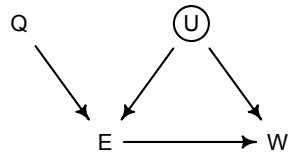
The back-door path $E \leftarrow U \rightarrow W$ ruins our day.

Even though we cannot condition on U, since we haven't observed it, there might be something we can do. If we can find a suitable **INSTRUMENTAL VARIABLE**. In causal terms, an instrumental variable is a variable that acts like a natural experiment on the exposure E. In technical terms, an instrumental variable Q is a variable that satisfies these criteria:

(1) Independent of U ($Q \perp\!\!\!\perp U$)
(2) Not independent of E ($Q \not\perp\!\!\!\perp E$)
(3) Q cannot influence W except through E

This last line is sometimes called the **EXCLUSION RESTRICTION**. It cannot be tested, and it is often implausible. Similarly, the first line above cannot be tested. But if you have a strong understanding of the system, so that you believe these criteria, then magic can happen.

It is much easier to understand instruments with a DAG. In our education and wages example, the simplest instrument for education looks like this:



The instrument here is Q. Given this DAG, Q satisfies all of the criteria for a valid instrumental variable. Note that valid instruments can be embedded in much more complicated graphs. If you can condition on other variables, in order to satisfy the criteria listed above, then you have an instrument.

How do we use Q in a model? You cannot just add it to a regression like any other predictor variable. Why not? Suppose we regress W on E. This is the relationship we'd like

to know. The association is however confounded by the back-door path through U. What happens if we then add Q to the model as another predictor? Bad stuff happens. There is no back-door path through Q, as you can see. But there is a non-causal path from Q to W through U: $Q \rightarrow E \leftarrow U \rightarrow W$. This is a non-causal path, because changing Q doesn't result in any change in W through this path. But since we are conditioning on E in the same model, and E is a collider of Q and U, the non-causal path is open. This confounds the coefficient on Q. It won't be zero, because it'll pick up the association between U and W. And then, as a result, the coefficient on E can get even more confounded. Used this way, an instrument like Q might be called a **BIAS AMPLIFIER**.[196]

This is all very confusing. Consider this example. Suppose Q indicates which quarter of the year—winter, spring, summer, fall—a person was born in. Why might this influence education? Because people born earlier in the year tend to get less schooling. This is both because they are biologically older when they start school and because they become eligible to drop out of school earlier. Now, if it is true that Q influences W only through E, and Q is also not influenced by confounds U, then Q is one of these mysterious instrumental variables. This means we can use it in a special way to make a valid causal inference about $E \rightarrow W$ without measuring U.

This example is based on a real study,[197] but let's simulate the data, both to keep it simple and to be sure what the right answer is. Remember: With real data, you never know what the right answer is. That is why studying simulated examples is so important, both for verifying that algorithms work and for schooling our intuition. Here are 500 simulated people:

R code 14.23

```
set.seed(73)
N <- 500
U_sim <- rnorm( N )
Q_sim <- sample( 1:4 , size=N , replace=TRUE )
E_sim <- rnorm( N , U_sim + Q_sim )
W_sim <- rnorm( N , U_sim + 0*E_sim )
dat_sim <- list(
    W=standardize(W_sim) ,
    E=standardize(E_sim) ,
    Q=standardize(Q_sim) )
```

The instrument Q varies from 1 to 4. Largest values are associated with more education, through the addition of `Q_sim` to the mean of `E_sim`. I've assumed that the true influence of education on wages is zero. This is just for the sake of the example. But the instrument Q does influence education, so it can serve as an instrument for discovering $E \rightarrow W$.

Let's consider three models. First, if we naively regress wages on education, the model will be confident that education causes higher wages:

R code 14.24

```
m14.4 <- ulam(
    alist(
        W ~ dnorm( mu , sigma ),
        mu <- aW + bEW*E,
        aW ~ dnorm( 0 , 0.2 ),
        bEW ~ dnorm( 0 , 0.5 ),
        sigma ~ dexp( 1 )
    ) , data=dat_sim , chains=4 , cores=4 )
```

```
precis( m14.4 )
```

```
      mean   sd  5.5% 94.5% n_eff Rhat
aW    0.00 0.04 -0.06  0.06  2024    1
bEW   0.40 0.04  0.33  0.46  1996    1
sigma 0.92 0.03  0.87  0.97  1861    1
```

This is just an ordinary confound, where the unmeasured U is ruining our inference. If you have incentives to believe that education enhances wages, you might report this inference as is. But even if E does increase W, the estimate from this model will be biased upwards. It's not enough to just know that E positively influences W. Accuracy matters.

Next let's consider what happens when we add Q as an ordinary predictor. Modifying the model above:

R code 14.25

```
m14.5 <- ulam(
    alist(
        W ~ dnorm( mu , sigma ),
        mu <- aW + bEW*E + bQW*Q,
        aW ~ dnorm( 0 , 0.2 ),
        bEW ~ dnorm( 0 , 0.5 ),
        bQW ~ dnorm( 0 , 0.5 ),
        sigma ~ dexp( 1 )
    ) , data=dat_sim , chains=4 , cores=4 )
precis( m14.5 )
```

```
       mean   sd  5.5% 94.5% n_eff Rhat
aW     0.00 0.04 -0.06  0.06  1526    1
bEW    0.64 0.05  0.56  0.71  1381    1
bQW   -0.41 0.05 -0.48 -0.33  1416    1
sigma  0.86 0.03  0.82  0.90  1823    1
```

This is a disaster. As expected from study of the DAG, `bQW` picks up an association from U. And `bEW` is even further from the truth now. It was 0.4 above. Now it's 0.64. That is bias amplification in action.

Now we're ready to see how to correctly use Q. The answer is actually pretty simple. We just use the generative model. Let's write a simple generative version of the DAG. It really has four sub-models. First, there is model for how wages W are caused by education E and the unobserved confound U. In mathematical notation:

$$W_i \sim \text{Normal}(\mu_{\text{W},i}, \sigma_{\text{W}})$$
$$\mu_{\text{W},i} = \alpha_{\text{W}} + \beta_{\text{EW}} E_i + U_i$$

Second, there is a model for how education levels E are caused by quarter of birth Q—this is our instrument recall—and the same unobserved confound U.

$$E_i \sim \text{Normal}(\mu_{\text{E},i}, \sigma_{\text{E}})$$
$$\mu_{\text{E},i} = \alpha_{\text{E}} + \beta_{\text{QE}} Q_i + U_i$$

The third model is for Q. The model just says that one-quarter of all people are born in each quarter of the year.

$$Q_i \sim \text{Categorical}([0.25, 0.25, 0.25, 0.25])$$

The fourth model says that the unobserved confound U is normally distributed with mean zero and standard deviation one.

$$U_i \sim \text{Normal}(0, 1)$$

U could have some other distribution. But this is the generative model at the moment.

Now we translate this generative model into a statistical model. We could do it brute force, just treating the U_i values as missing data and imputing them. But you won't see how to do that until the next chapter. Besides, it is much more efficient to instead average over them and estimate instead the covariance between W and E. That's what we'll do: Define W and E as coming from a common multivariate normal distribution. Like this:

$$\begin{pmatrix} W_i \\ E_i \end{pmatrix} \sim \text{MVNormal}\left(\begin{pmatrix} \mu_{\text{W},i} \\ \mu_{\text{E},i} \end{pmatrix}, \mathbf{S} \right) \qquad \text{[Joint wage \& education model]}$$
$$\mu_{\text{W},i} = \alpha_{\text{W}} + \beta_{\text{EW}} E_i$$
$$\mu_{\text{E},i} = \alpha_{\text{E}} + \beta_{\text{QE}} Q_i$$

The matrix $\mathbf{S}$ in the first line is the error covariance between wages and education. It's not the descriptive covariance between these variables, but rather the matrix equivalent of the typical σ we stick in a Gaussian regression. The above is a **MULTIVARIATE LINEAR MODEL**, a regression with multiple simultaneous outcomes, all modeled with a joint error structure. Each variable gets its own linear model, yielding the two μ definitions. It might bother you to see education E as both an outcome and a predictor inside the mean for W. But this statistical relationship is an implication of the DAG. There is nothing illegal about it. All it says is that E might influence W and that also pairs of W, E values might have some residual correlation. That correlation arises, presuming the DAG, through the unobserved confound U.

The full model also needs priors, of course. We standardized the variables, so we can use our default priors for standardized linear regression. Here's the `ulam` code:

R code 14.26

```
m14.6 <- ulam(
    alist(
        c(W,E) ~ multi_normal( c(muW,muE) , Rho , Sigma ),
        muW <- aW + bEW*E,
        muE <- aE + bQE*Q,
        c(aW,aE) ~ normal( 0 , 0.2 ),
        c(bEW,bQE) ~ normal( 0 , 0.5 ),
        Rho ~ lkj_corr( 2 ),
        Sigma ~ exponential( 1 )
    ), data=dat_sim , chains=4 , cores=4 )
precis( m14.6 , depth=3 )
```

```
           mean   sd  5.5% 94.5% n_eff Rhat
aE         0.00 0.03 -0.06  0.05  1351    1
aW         0.00 0.04 -0.07  0.07  1432    1
bQE        0.59 0.04  0.53  0.64  1321    1
bEW       -0.05 0.08 -0.18  0.07  1024    1
Rho[1,1]   1.00 0.00  1.00  1.00   NaN  NaN
Rho[1,2]   0.54 0.05  0.46  0.62  1080    1
Rho[2,1]   0.54 0.05  0.46  0.62  1080    1
Rho[2,2]   1.00 0.00  1.00  1.00  1361    1
Sigma[1]   1.02 0.05  0.95  1.10  1085    1
```

```
Sigma[2]  0.81 0.02  0.77  0.85  1768    1
```

There is a lot going on here. But we can take it one piece at a time. First look at `bEW`, the estimated influence of education on wages. It is small and straddles both sides of zero. That is the correct causal inference. Second, the correlation `Rho[1,2]` between the two outcomes, wages and education, is reliably positive. That reflects the common influence of U. Remember: This correlation is conditional on E (for W) and Q (for E). It isn't the raw empirical correlation, but rather the residual correlation.

It's a good idea to adjust the simulation and try other scenarios. To speed up your play, you can avoid re-compiling the models as long as you keep `N=500` and run these lines to sample from the posterior distributions:

R code 14.27

```
m14.4x <- ulam( m14.4 , data=dat_sim , chains=4 , cores=4 )
m14.6x <- ulam( m14.6 , data=dat_sim , chains=4 , cores=4 )
```

To begin, you might try a scenario in which education has a positive influence but the confound hides it:

R code 14.28

```
set.seed(73)
N <- 500
U_sim <- rnorm( N )
Q_sim <- sample( 1:4 , size=N , replace=TRUE )
E_sim <- rnorm( N , U_sim + Q_sim )
W_sim <- rnorm( N , -U_sim + 0.2*E_sim )
dat_sim <- list(
    W=standardize(W_sim) ,
    E=standardize(E_sim) ,
    Q=standardize(Q_sim) )
```

You should find that E and W have a negative correlation in their residual variance, because the confound positively influences one and negatively influences the other.

Instrumental variables are hard to understand. But there are some excellent tools to help you. For example, the `dagitty` package contains a function `instrumentalVariables` that will find instruments, if they are present in a DAG. In this example, we could define the DAG and query the instrument this way:

R code 14.29

```
library(dagitty)
dagIV <- dagitty( "dag{ Q -> E <- U -> W <- E }" )
instrumentalVariables( dagIV , exposure="E" , outcome="W" )
```

```
 Q
```

This is no substitute for understanding, but it can help you develop understanding and check your intuitions.

The hardest thing about instrumental variables is believing in any particular instrument. If you believe in your DAG, they can are easy to believe. But should you believe in your DAG? As an example, a study of islands employed wind direction as an instrument for inferring the impact of colonialism on economic development.[198] Colonial history and economic performance are confounded by many things, like the natural resources of an island. If however wind direction influences date of colonization—because when ships used sails, trade winds

made some islands easier to reach—but not economic performance directly, then it could serve as an instrument. This is a very clever idea. But it is easy to imagine that wind influences many things about an island, including its pre-colonial history of contact and its ecology, and that these variables will influence current economies.

A much more common type of instrument is distance to some service. If for example we want to estimate the influence of health care on the wellbeing of mothers, we cannot easily randomize health care among mothers. It would be unethical, for starters. But if mothers naturally vary in distance to care centers, and these distances are random with respect to pre-existing health variables, then distance might be an instrument that influences use of health care but does not influence health directly. However. it's not hard to think of ways that distance from a hospital could be associated with factors influencing health, violating the exclusion restriction.[199]

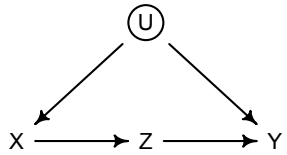
In general, it is not possible to statistical prove whether a variable is a good instrument. As always, we need scientific knowledge outside of the data to make sense of the data.

Rethinking: Two-stage worst squares. The instrumental variable model is often discussed with an estimation procedure known as TWO-STAGE LEAST SQUARES (2SLS). This procedure involves two linear regressions. The predicted values of the first regression are fed into the second as data, and then adjustments are made so that the standard errors make sense. Amazingly, when the weather is nice, this procedure works. It relies upon large-sample approximations and has well-known problems.[200] Like all golems, you just have to use it responsibly. Sometimes people mistake the procedure of 2SLS for the model of instrumental variables. They are not the same thing. Any model can be estimated through a number of different procedures, each with its own benefits and costs. 2SLS is very limiting. If we have count outcomes, measurement errors, missing values, or need varying effects, 2SLS is dubious. Now that more capable procedures exist, it is easier to fit instrumental variable models. But it can still be difficult. There are no guarantees that an effect can be estimated, just because the DAG says it is possible. Another issue that will always remain, no matter how you approximate the posterior, is that it is very hard to be sure the instrumental variable is any good.

14.3.2. Other designs. Instrumental variables are natural experiments that impersonate randomized experiments. In the example in the previous section, quarter of birth Q is like an external manipulation of education E. That external shock to education is like an experimental manipulation, in the sense that it allows us to estimate the impact of that external shock and thereby derive a causal estimate.

There are potentially many ways to find natural experiments. Not all of them are strictly instrumental variables. But they can provide theoretically correct designs for causal inference, if you can believe the assumptions. Let's consider two more.

In addition to the back-door criterion you met in Chapter 6, there is something called the FRONT-DOOR CRITERION. It is relevant in a DAG like this:



We are interest, as usual, in the causal influence of X on Y. But there is an unobserved confound U, again as usual. It turns out that, if we can find a perfect mediator Z, then we can possibly estimate the causal effect of X on Y. It isn't crazy to think that causes are mediated

by other causes. Everything has a mechanism. Z in the DAG above is such a mechanism. If you have a believable Z variable, then the causal effect of X on Y is estimated by expressing the generative model as a statistical model, similar to the instrumental variable example before. In special cases, such as when everything is linear and Gaussian, there is a formula. But we don't need formulas. We just need to think generatively and use Bayes.

The front-door criterion isn't used much. This may be because it is relatively new or rather that believable Z variables are rare. A possible example is the influence of social ties formed in college on voting behavior in the United States Senate.[201] The question is whether senators who went to the same college vote more similarly, because their social ties produce coordinated votes. The pure association between attending the same college and voting the same way is obviously confounded by lots of things. The front-door trick is to find some mechanism through which social ties must act. In the case of the United States Senate, a mechanism could be who sits next to who. It is easier to talk to and coordinate with people sitting nearby. And since junior members are often assigned seats effectively at random, seating is unlikely to share the same confounds as college attendance. Now consider some senators who attended UCLA. Some of them end up seated near one another. Others end up seated next to dreadful UC Berkeley alums. If the ones seated near one another vote more similarly to one another than to the UCLA alums seated elsewhere, that could be causal evidence that social ties influence voting, as mediated by proximity on the Senate floor.

A much more common design is **REGRESSION DISCONTINUITY** (or **RDD**). This is really a special kind of instrumental variable design. Suppose for example that we want to estimate the effect of winning an academic award on future success.[202] This is confounded by unobserved factors, like ability, that influence both the award and later success. But if we compare individuals who were just below the cutoff for the award to those who were just above the cutoff, these individuals should be more similar in the unobserved factors. It's as if the award is applied at random, for individuals close to the cutoff. This is key idea behind regression discontinuity. In practice, one regression is fit for individuals above the cutoff and another to those below the cutoff. Then an estimate of the causal effect is the average difference between individuals just above and just below the cutoff. While the difference near the cutoff is of interest, the entire function influences this difference. So some care is needed in choosing functions for the overall relationship between the exposure and the outcome.[203]

14.4. Social relations as correlated varying effects

Once you grasp the basic strategy of using covariance matrixes to represent populations of correlated effects, you can accomplish a lot of different and scientifically relevant modeling goals. In this section, I present an example that constructs a custom covariance matrix with special scientific meaning.

The data we'll work with are `data(KosterLeckie)`, which loads two different tables, `kl_dyads` and `kl_households`. See `?KosterLeckie` for more details.[204]

R code 14.30

```
library(rethinking)
data(KosterLeckie)
```

For now, we want to use the variables in `kl_dyads`. Each row in this table is a dyad of households from a community in Nicaragua. We are interested in modeling gift exchanges among these households. The outcome variables `giftsAB` and `giftsBA` in each row are the count of gifts in each direction within each dyad. The variables `hidA` and `hidB` tell us the

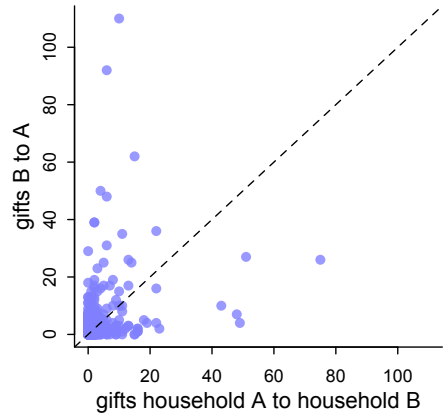


FIGURE 14.8. Distribution of dyadic gifts in `data(KosterLeckie)`. 25 households present 300 dyads, with an overall correlation of 0.24. But to get a sensible measure of balance of gift giving, we need to make a model that deals with the repeat presence of specific households across dyads.

household IDs in each dyad, and `did` is a unique dyad ID number. We'll ignore the other variables for now.

FIGURE 14.8 shows the raw distribution of gifts across dyads. The overall correlation here is 0.24. But taking this as a measure of balance of exchange would be a bad idea. First, the correlation changes if we switch the A/B labels. Since the labels are arbitrary, that means the measured correlation is also somewhat arbitrary. Second, the generative model in the background is that gifts can be explained both by the special relationship in each dyad—some households tend to exchange gifts frequently—as well as by the fact that some households give or receive a lot across all dyads, without regard to any special relationships among households. For example, if a household is poor, it might not give many gifts, but it might receive many. In order to statistically separate balanced exchange from generalized differences in giving and receiving, we need a model that treats these as separate. The type of model we'll consider is often called a **SOCIAL RELATIONS MODEL**, or SRM.

Specifically, we'll model gifts from household A to household B as a combination of varying effects specific to the household and the dyad. The outcome variables, the gift counts, are Poisson variables—they are counts with no obvious upper bound. We'll attach our varying effects to these counts with a log link, as in the previous chapters. This gives us the first part of the model:

$$\begin{aligned} y_{A\to B} &\sim \text{Poisson}(\lambda_{AB}) \\ \log \lambda_{AB} &= \alpha + g_A + r_B + d_{AB} \end{aligned}$$

The linear model has an intercept α that represent the average gifting rate (on the log scale) across all dyads. The other effects will be offsets from this average. Then g_A is a varying effect parameter for the generalized giving tendency of household A, regardless of dyad. The effect r_B is the generalized receiving of household B, regardless of dyad. Finally the effect d_{AB} is the dyad-specific rate that A gives to B. There is a corresponding linear model for the other direction within the same dyad:

$$\begin{aligned} y_{B\to A} &\sim \text{Poisson}(\lambda_{BA}) \\ \log \lambda_{BA} &= \alpha + g_B + r_A + d_{BA} \end{aligned}$$

Together, this all implies that each household H needs varying effects, a g_H and a r_H. In addition each dyad AB has two varying effects, d_{AB} and d_{BA}. We want to allow the g and r parameters to be correlated—do people who give a lot also get a lot? We also want to allow the dyad effects to be correlated—is there balance within dyads? We can do all of this with two difference multi-normal priors. The first will represent the population of household effects:

$$\begin{pmatrix} g_i \\ r_i \end{pmatrix} \sim \text{MVNormal}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_g^2 & \sigma_g\sigma_r\rho_{gr} \\ \sigma_g\sigma_r\rho_{gr} & \sigma_r^2 \end{pmatrix}\right)$$

For any household i, a pair of g and r parameters are assigned a prior with a typical covariance matrix with two standard deviations and a correlation parameter. There's nothing new here.

The second multi-normal prior will represent the population of dyad effects:

$$\begin{pmatrix} d_{ij} \\ d_{ji} \end{pmatrix} \sim \text{MVNormal}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_d^2 & \sigma_d^2\rho_d \\ \sigma_d^2\rho_d & \sigma_d^2 \end{pmatrix}\right)$$

For a dyad with households i and j, there is a pair of dyad effects with a prior with another covariance matrix. But this matrix is funny. Take a close look and you'll see that there is only one standard deviation parameter, σ_d. Why? Because the labels in each dyad are arbitrary. It isn't meaningful which household comes first or second. So both parameters must have the same variance. But we do want to estimate their correlation, and that is what ρ_d will do for us. If ρ_d is positive, then when one household gives more within a dyad, so too does the other. If ρ_d is negative, then when one households gives more, the other gives less. If ρ_d is instead near zero, then there is no pattern within dyads.

Let's build this model now. We need to construct the dyad covariance matrix in a custom way, and we need to be careful with indexing the varying effects. Here is the model:

R code 14.31

```
kl_data <- list(
    N = nrow(kl_dyads),
    N_households = max(kl_dyads$hidB),
    did = kl_dyads$did,
    hidA = kl_dyads$hidA,
    hidB = kl_dyads$hidB,
    giftsAB = kl_dyads$giftsAB,
    giftsBA = kl_dyads$giftsBA
)

m14.7 <- ulam(
    alist(
        giftsAB ~ poisson( lambdaAB ),
        giftsBA ~ poisson( lambdaBA ),
        log(lambdaAB) <- a + gr[hidA,1] + gr[hidB,2] + d[did,1] ,
        log(lambdaBA) <- a + gr[hidB,1] + gr[hidA,2] + d[did,2] ,
        a ~ normal(0,1),

       ## gr matrix of varying effects
        vector[2]:gr[N_households] ~ multi_normal(0,Rho_gr,sigma_gr),
        Rho_gr ~ lkj_corr(4),
        sigma_gr ~ exponential(1),

       ## dyad effects
```

```
        transpars> matrix[N,2]:d <-
                compose_noncentered( rep_vector(sigma_d,2) , L_Rho_d , z ),
        matrix[2,N]:z ~ normal( 0 , 1 ),
        cholesky_factor_corr[2]:L_Rho_d ~ lkj_corr_cholesky( 8 ),
        sigma_d ~ exponential(1),

       ## compute correlation matrix for dyads
        gq> matrix[2,2]:Rho_d <<- Chol_to_Corr( L_Rho_d )
    ), data=kl_data , chains=4 , cores=4 , iter=2000 )
```

I've broken this up into sections, to make it easier to read. The top section is the two outcome variables, each direction of gifting in the dyad. Each linear model contains the intercept `a`. Then comes a giving effect for the household giving on that line, `gr[hidA,1]` or `gr[hidB,1]`. That "1" is for the first column of the `gr` matrix. Then comes the receiving effect for the household receiving, either `gr[hidB,2]` or `gr[hidA,2]`. Finally, the dyad effects `d[did,1]` for household A and `d[did,2]` for household B. This is because we put household A in the first column of the `d` matrix. The order is arbitrary, since A and B are just labels.

The next chunk of code defines the matrix of giving and receiving effects. The matrix `gr` will have a row for each household and 2 columns. The first column will be the giving varying effect and the second column will be the receiving varying effect, just like in the linear models.

The third chunk defines the special dyad matrix. These are non-centered, for the sake of efficient mixing. The special piece is the `rep_vector(sigma_d,2)`. This copies the standard deviation into a vector of length 2 and composes the covariance matrix from there. So we end up with the correct covariance matrix, with the same variance for both effects.

Finally, there is a single line at the bottom that computes the correlation matrix for the dyads. This is necessary, because the model is parameterized using a Cholesky factor. The function `Chol_to_Corr` multiplies a matrix by its own transpose. This is how a Cholesky factor is made back into its original matrix. If you want to interpret the correlations among the effects, then this is a useful calculation. The `gq>` at the start of the line places the line in Stan's generated quantities block, which holds code that is executed after each Hamiltonian transition. So anything you want calculated from each sample should be tagged in this way. It will show up in the posterior distribution.

This model contains a lot of parameters. There are 600 dyad parameters, for example. But we can get some useful information from the covariance matrix components:

R code 14.32

```
precis( m14.7 , depth=3 , pars=c("Rho_gr","sigma_gr") )
```

```
             mean   sd  5.5% 94.5% n_eff Rhat
Rho_gr[1,1]  1.00 0.00  1.00  1.00   NaN  NaN
Rho_gr[1,2] -0.40 0.20 -0.70 -0.07  1475    1
Rho_gr[2,1] -0.40 0.20 -0.70 -0.07  1475    1
Rho_gr[2,2]  1.00 0.00  1.00  1.00  3834    1
sigma_gr[1]  0.83 0.14  0.64  1.07  2371    1
sigma_gr[2]  0.42 0.09  0.29  0.57  1251    1
```

As in other models with covariance matrixes, since the diagonal cells are always 1, you can ignore those lines in the output. The parameters `Rho_gr[1,2]` and `Rho_gr[2,1]` are actually the same parameter, because the matrix is symmetric. The correlated between general giving and receiving is negative, with an 89% compatibility interval from about -0.7 to -0.1. This implies that individuals who give more across all dyads tend to receive less. The standard deviation parameters `sigma_gr[1]` and `sigma_gr[2]` show clear evidence that rates of giving are more variable than rates of receiving.

Let's plot these giving and receiving effects, so you can see this covariance structure in the parameters. We want to calculate, for each household, its posterior predictive giving and receiving rates, across all dyads. We can do this by using the linear model directly to add the intercept `a` to each giving or receiving parameter:

R code 14.33

```
post <- extract.samples( m14.7 )
g <- sapply( 1:25 , function(i) post$a + post$gr[,i,1] )
r <- sapply( 1:25 , function(i) post$a + post$gr[,i,2] )
Eg_mu <- apply( exp(g) , 2 , mean )
Er_mu <- apply( exp(r) , 2 , mean )
```

If you look at `str(g)`, you'll see a matrix with 4000 rows (samples) and 25 columns (households). These are the posterior distributions of giving for each household. The matrix `r` is the same for receiving. `Eg_mu` and `Er_mu` holds the means on the outcome scale. That's why they were exponentiated.

Before plotting those points, I'd like to also show the uncertainty around each. How can we do that? There is uncertainty in both directions, because there is a distribution with some correlation structure here. We could just plot the columns in `g` and `r`. Try `plot(exp(g[,1]),exp(r[,1]))` for example to show the posterior distribution of giving/receiving for household number 1. That is messy, but it does show the uncertainty in each household's values.

We can produce a cleaner visualization with some contours. On the latent scale of the linear model, the bivariate distribution of each `g` and `r` is approximately Gaussian. So we can describe its shape with an ellipse. If we then project this ellipse onto the outcome scale, we'll have a clean contour for the uncertainty.

R code 14.34

```
plot( NULL , xlim=c(0,8.6) , ylim=c(0,8.6) , xlab="generalized giving" ,
    ylab="generalized receiving" , lwd=1.5 )
abline(a=0,b=1,lty=2)

# ellipses
library(ellipse)
for ( i in 1:25 ) {
    Sigma <- cov( cbind( g[,i] , r[,i] ) )
    Mu <- c( mean(g[,i]) , mean(r[,i]) )
    for ( l in c(0.5) ) {
        el <- ellipse( Sigma , centre=Mu , level=l )
        lines( exp(el) , col=col.alpha("black",0.5) )
    }
}
# household means
```

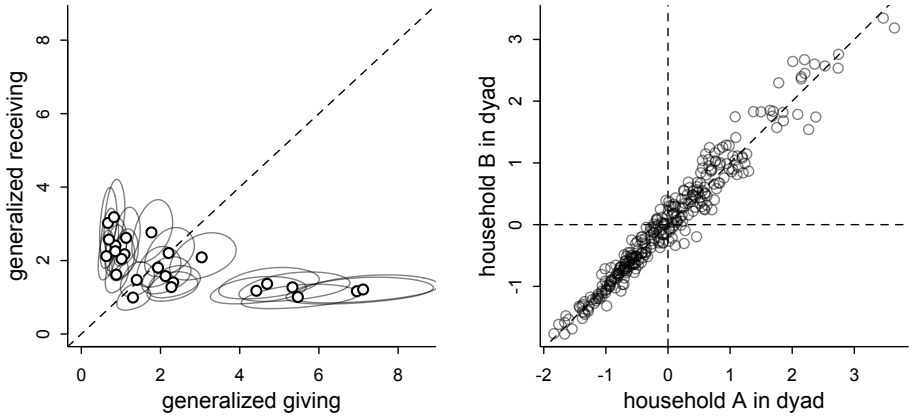


FIGURE 14.9. Left: Expected giving and receiving, absent any dyad-specific effects. Each point is a household and the ellipses show 50% compatibility regions. There is a negative relationship between average giving and average receiving across households. Right: Dyad-specific effects, absent generalized giving and receiving. After accounting for overall rates of giving and receiving, residual gifts are strongly correlated within dyads.

```
points( Eg_mu , Er_mu , pch=21 , bg="white" , lwd=1.5 )
```

The left side of FIGURE 14.9 shows the result. Note the negative relationship between giving on the horizontal and receiving on the vertical. The dashed line shows where the two rates would be equal. The households with the lowest rates of giving have some of the highest rates of receiving. This likely reflects need-based gifts. Likewise the households with the highest rates of giving have some of the lowest rates of receiving. That is the negative correlation we saw in the `precis` output. Note also the greater variation in giving rates. That corresponds to the standard deviation parameters.

Now what about the dyad effects? Let's look at that covariance matrix:

R code 14.35

```
precis( m14.7 , depth=3 , pars=c("Rho_d","sigma_d") )
```

```
           mean   sd 5.5% 94.5% n_eff Rhat
Rho_d[1,1] 1.00 0.00 1.00  1.00   NaN  NaN
Rho_d[1,2] 0.88 0.03 0.83  0.93  1287    1
Rho_d[2,1] 0.88 0.03 0.83  0.93  1287    1
Rho_d[2,2] 1.00 0.00 1.00  1.00   NaN  NaN
sigma_d    1.11 0.06 1.02  1.20  1583    1
```

The correlation here is positive and strong. And there is more variation among dyads than there is among household in giving rates. This implies that pairs of households are balanced—if one household gives less than average (after accounting for generalized giving and receiving), then the other probably gives less as well. We can plot the raw dyad effects to see how strong this pattern is:

R code 14.36

```
dy1 <- apply( post$d[,,1] , 2 , mean )
dy2 <- apply( post$d[,,2] , 2 , mean )
plot( dy1 , dy2 )
```

The result is the righthand plot in FIGURE 14.9. These are only posterior means—there is a lot of uncertainty about each dyad. But there is an astonishing amount of balance. This could reflect reciprocity, adjusted for overall wealth levels. Or it could reflect types of relationships among households, like kin obligations, that we haven't included in the model.

The full dataset contains a number of covariates that can be used to explain these effects: economic activities, relationships, distances among households. A model like this one, with only varying effects, can partition the variation and show us where the action is. But our goal is to gain some causal understanding through adding more information to the model.

Rethinking: Where everybody knows your name. The gift example is a SOCIAL NETWORK model. In that light, an important feature missing from this model is the TRANSITIVITY of social relationships. If household A is friends with household B, and household C is friends with household B, then households A and C are more likely to be friends. This isn't magic. It just arises from unobserved factors that create correlated relationships. For example, people who go to the same pub tend to know one another. The pub is an unmeasured confound for inferring causes of social relations. Models that can estimate and expect transitivity can be better. This can be done using something called a STOCHASTIC BLOCK MODEL. However to fit such a model, we'll need some techniques in the next chapter.

14.5. Continuous categories and the Gaussian process

All of the varying effects so far, whether they were intercepts or slopes, have been defined over discrete, unordered categories. For example, cafés are unique places, and there is no sense in which café 1 comes before café 2. The "1" and "2" are just labels for unique things. The same goes for tadpole ponds, academic departments, or individual chimpanzees. By estimating unique parameters for each cluster of this kind, we can quantify some of the unique features that generate variation across clusters and covariation among the observations within each cluster. Pooling across the clusters improves accuracy and simultaneously provides a picture of the variation.

But what about continuous dimensions of variation like age or income or stature? Individuals of the same age share some of the same exposures. They listened to some of the same music, heard about the same politicians, and experienced the same weather events. And individuals of *similar* ages also experienced some of these same exposures, but to a lesser extent than individuals of the same age. The covariation falls off as any two individuals become increasingly dissimilar in age or income or stature or any other dimension that indexes background similarity. It doesn't make sense to estimate a unique varying intercept for all individuals of the same age, ignoring the fact that individuals of similar ages should have more similar intercepts. And of course, it's likely that every individual in your sample has a unique age. So then continuous differences in similarity are all you have to work with.

Luckily, there is a way to apply the varying effects approach to continuous categories of this kind. This will allow us to estimate a unique intercept (or slope) for any age, while still regarding age as a continuous dimension in which similar ages have more similar intercepts (or slopes). The general approach is known as GAUSSIAN PROCESS REGRESSION.[205] This name is unfortunately wholly uninformative about what it is for and how it works.

We'll proceed to work through a basic example that demonstrates both what it is for and how it works. The general purpose is to define some dimension along which cases differ. This might be individual differences in age. Or it could be differences in location. Then we measure the distance between each pair of cases. What the model then does is estimate a function for the covariance between pairs of cases at different distances. This covariance function provides one continuous category generalization of the varying effects approach.

14.5.1. Example: Spatial autocorrelation in Oceanic tools. When we looked at the complexity of tool kits among historic Oceanic societies, back in Chapter 11 (page 354), we used a crude binary contact predictor as a proxy for possible exchange among societies. But that variable is pretty unsatisfying. First, it takes no note of which other societies each had contact (or not) with. If all of your neighbors are small islands, then high rate of contact with them may not do much at all to tool complexity. Second, if indeed tools were exchanged among societies—and we know they were—then the total number of tools for each are truly not independent of one another, even after we condition on all of the predictors. Instead we expect close geographic neighbors to have more similar tool counts, because of exchange. Third, closer islands may share unmeasured geographic features like sources of stone or shell that lead to similar technological industries. So space could matter in multiple ways.

This is a classic setting in which to use Gaussian process regression. We'll define a distance matrix among the societies. Then we can estimate how similarity in tool counts depends upon geographic distance. You'll see how to simultaneously incorporate ordinary predictors, so that the covariation among societies with distance will both control for and be controlled by other factors that influence technology.

Let's begin by loading the data and inspecting the geographic distance matrix. I've already gone ahead and looked up the as-the-crow-flies navigation distance between each pair of societies. These distances are measured in thousands of kilometers, and the matrix of them is in the `rethinking` package:

R code 14.37

```
# load the distance matrix
library(rethinking)
data(islandsDistMatrix)

# display (measured in thousands of km)
Dmat <- islandsDistMatrix
colnames(Dmat) <- c("Ml","Ti","SC","Ya","Fi","Tr","Ch","Mn","To","Ha")
round(Dmat,1)
```

```
           Ml  Ti  SC  Ya  Fi  Tr  Ch  Mn  To  Ha
Malekula   0.0 0.5 0.6 4.4 1.2 2.0 3.2 2.8 1.9 5.7
Tikopia    0.5 0.0 0.3 4.2 1.2 2.0 2.9 2.7 2.0 5.3
Santa Cruz 0.6 0.3 0.0 3.9 1.6 1.7 2.6 2.4 2.3 5.4
Yap        4.4 4.2 3.9 0.0 5.4 2.5 1.6 1.6 6.1 7.2
Lau Fiji   1.2 1.2 1.6 5.4 0.0 3.2 4.0 3.9 0.8 4.9
Trobriand  2.0 2.0 1.7 2.5 3.2 0.0 1.8 0.8 3.9 6.7
Chuuk      3.2 2.9 2.6 1.6 4.0 1.8 0.0 1.2 4.8 5.8
Manus      2.8 2.7 2.4 1.6 3.9 0.8 1.2 0.0 4.6 6.7
Tonga      1.9 2.0 2.3 6.1 0.8 3.9 4.8 4.6 0.0 5.0
Hawaii     5.7 5.3 5.4 7.2 4.9 6.7 5.8 6.7 5.0 0.0
```

Notice that the diagonal is all zeros, because each society is zero kilometers from itself. Also notice that the matrix is symmetric around the diagonal, because the distance between two societies is the same whichever society we measure from.

We'll use these distances as a measure of similarity in technology exposure. This will allow us to estimate varying intercepts for each society that account for non-independence in tools as a function of their geographical similarly. The notion is that the expected number of tools for each society gets a varying intercept, based on a continuous distance measure, that makes it correlated with the tool counts of its neighbors.

We'll use the "scientific" tool model from Chapter 11. In that model, the first part of the model is a familiar Poisson probably of the outcome variable. Then there is a model-derived expected number of tools::

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

We'd like to have these λ values adjusted by a varying intercept parameter. We could just add the intercept to the expression above, but then λ_i might end up negative. So instead let's make the varying intercepts multiplicative:

$$\begin{aligned} T_i &\sim \text{Poisson}(\lambda_i) \\ \lambda_i &= \exp(k_{\text{SOCIETY}[i]})\alpha P_i^{\beta}/\gamma \end{aligned}$$

where $k_{\text{SOCIETY}[i]}$ is the varying intercept. But unlike typical varying intercepts, it will be estimated in light of geographic distance, not distinct category membership.

The heart of the Gaussian process is the multivariate prior for these intercepts:

$$\begin{pmatrix} k_1 \\ k_2 \\ k_3 \\ \dots \\ k_{10} \end{pmatrix} \sim \text{MVNormal}\left(\begin{pmatrix} 0 \\ 0 \\ 0 \\ \dots \\ 0 \end{pmatrix}, \mathbf{K}\right) \quad \text{[prior for intercepts]}$$

$$\mathbf{K}_{ij} = \eta^2 \exp(-\rho^2 D_{ij}^2) + \delta_{ij}\sigma^2 \quad \text{[define covariance matrix]}$$

The first line is the 10-dimensional Gaussian prior for the intercepts. It has 10 dimensions, because there are 10 societies in the distance matrix. The vector of means is all zeros, which means the inside the linear model the average society will multiply λ by $\exp(0) = 1$. So the average doesn't change the expectation. Negative k values will reduce λ, and positive k values will increase it.

The covariance matrix for these intercepts is named $\mathbf{K}$, and the covariance between any pair of societies i and j is $\mathbf{K}_{ij}$. This covariance is defined by the formula on the second line above. This formula uses three parameters—η, ρ, and σ—to model how covariance among societies changes with distances among them. It probably looks very unfamiliar. I'll walk you through it in pieces.

The part of the formula for $\mathbf{K}$ that gives the covariance model its shape is $\exp(-\rho^2 D_{ij}^2)$. D_{ij} is the distance between the i-th and j-th societies. So what this function says is that the covariance between any two societies i and j declines exponentially with the squared distance between them. The parameter ρ determines the rate of decline. If it is large, then covariance declines rapidly with squared distance.

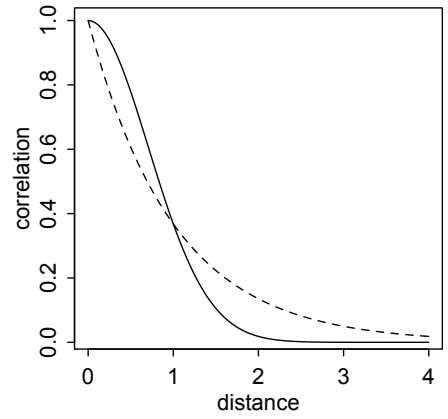


FIGURE 14.10. Shape of the function relating distance to the covariance $\mathbf{K}_{ij}$. The horizontal axis is distance. The vertical is the correlation, relative to maximum, between any two societies i and j. The dashed curve is the linear distance function. The solid curve is the squared distance function.

Why square the distance? You don't have to. This is just a model. But the squared distance is the most common assumption, both because it is easy to fit to data and has the often-realistic property of allowing covariance to decline more quickly as distance grows. This will be easy to appreciate, if we plot this function under the linear-decline alternative, $\exp(-\rho^2 D_{ij})$, and compare. We'll use a value $\rho^2 = 1$, just for the example.

R code 14.38

```
# linear
curve( exp(-1*x) , from=0 , to=4 , lty=2 ,
    xlab="distance" , ylab="correlation" )

# squared
curve( exp(-1*x^2) , add=TRUE )
```

The result is shown in FIGURE 14.10. The vertical axis here is just part of the total covariance function. You can think of it as the proportion of the maximum correlation between two societies i and j. The dashed curve is the linear distance function. It produces an exact exponential shape. The solid curve is the squared distance function. It produces a half-Gaussian decline that is initially slower than the exponential but rapidly accelerates and then becomes faster than exponential.

The last two pieces of $\mathbf{K}_{ij}$ are simpler. η^2 is the maximum covariance between any two societies i and j. The term on the end, $\delta_{ij}\sigma^2$, provides for extra covariance beyond η^2 when $i = j$. It does this because the function δ_{ij} is equal to 1 when $i = j$ but is zero otherwise. In the Oceanic societies data, this term will not matter, because we only have one observation for each society. But if we had more than one observation per society, σ here describes how these observations covary.

The model computes the posterior distribution of ρ, η, and σ. But it also needs priors for them. We'll define priors for the square of each, and estimate them on the same scale, because that's computationally easier. We don't need σ in this model, so we'll instead just fix it at an irrelevant constant.

Now here's the full model, with the fixed priors for each parameter added at the bottom:

$$
\begin{aligned}
T_i &\sim \text{Poisson}(\lambda_i) \\
\lambda_i &= \exp(k_{\text{SOCIETY}[i]})\alpha P_i^{\beta}/\gamma \\
\mathbf{k} &\sim \text{MVNormal}\big((0,\ldots,0),\mathbf{K}\big) \\
\mathbf{K}_{ij} &= \eta^2 \exp(-\rho^2 D_{ij}^2) + \delta_{ij}(0.01) \\
\alpha &\sim \text{Exponential}(1) \\
\beta &\sim \text{Exponential}(1) \\
\eta^2 &\sim \text{Exponential}(2) \\
\rho^2 &\sim \text{Exponential}(0.5)
\end{aligned}
$$

Note that ρ^2 and η^2 must be positive, so we place exponential priors on them. A little knowledge of Pacific navigation would probably allow us a smart, informative prior on ρ^2 at least.

We're finally ready to fit the model. The distribution to use, so to signal to `ulam` that you want to the squared distance Gaussian process prior, is `GPL2`. The rest of the code should be familiar.

R code 14.39

```
data(Kline2) # load the ordinary data, now with coordinates
d <- Kline2
d$society <- 1:10 # index observations

dat_list <- list(
    T = d$total_tools,
    P = d$population,
    society = d$society,
    Dmat=islandsDistMatrix )

m14.8 <- ulam(
    alist(
        T ~ dpois(lambda),
        lambda <- (a*P^b/g)*exp(k[society]),
        vector[10]:k ~ multi_normal( 0 , SIGMA ),
        matrix[10,10]:SIGMA <- cov_GPL2( Dmat , etasq , rhosq , 0.01 ),
        c(a,b,g) ~ dexp( 1 ),
        etasq ~ dexp( 2 ),
        rhosq ~ dexp( 0.5 )
    ), data=dat_list , chains=4 , cores=4 , iter=2000 )
```

Be sure to check the chains. They should sample well, but we could also improve sampling by de-centering the prior for k. We'll do that in box further down. Let's check the posterior:

R code 14.40

```
precis( m14.8 , depth=3 )
```

```
       mean   sd  5.5% 94.5% n_eff Rhat
k[1]  -0.17 0.30 -0.65  0.29   714 1.00
k[2]  -0.03 0.29 -0.48  0.43   538 1.01
k[3]  -0.08 0.28 -0.51  0.35   527 1.01
```

```
k[4]   0.34 0.26 -0.04  0.74   593 1.01
k[5]   0.07 0.25 -0.32  0.46   590 1.01
k[6]  -0.39 0.27 -0.84  0.00   789 1.00
k[7]   0.13 0.25 -0.26  0.53   606 1.01
k[8]  -0.22 0.26 -0.64  0.16   726 1.01
k[9]   0.26 0.25 -0.11  0.64   668 1.01
k[10] -0.18 0.35 -0.75  0.35   868 1.01
g      0.60 0.56  0.08  1.68  1536 1.00
b      0.28 0.08  0.15  0.41  1107 1.00
a      1.41 1.08  0.24  3.39  1811 1.00
etasq  0.20 0.20  0.03  0.56   863 1.00
rhosq  1.31 1.60  0.08  4.41  1931 1.00
```

First, note that the coefficient for log population, `bp`, is very much as it was before we added all this Gaussian process stuff. This suggests that it's hard to explain all of the association between tool counts and population as a side effect of geographic contact. Second, those `g` parameters are the Gaussian process varying intercepts for each society. Like `a` and `bp`, they are on the log-count scale, so they are hard to interpret raw.

In order to understand the parameters that describe the covariance with distance, `rhosq` and `etasq`, we'll want to plot the function they imply. Actually the joint posterior distribution of these two parameters defines a posterior distribution of covariance functions. We can get a sense of this distribution of functions—I know, this is rather meta—by plotting a bunch of them. Here we'll sample 100 from the posterior and display them along with the posterior median. Why use the median? Because the densities for `rhosq` and `etasq` are skewed. You can detect this in the `precis` output above: the mean for `rhosq` isn't even inside the 89% HPDI. So the median is a better measure of the center of mass than the mean. But as always, it is the entire distribution that matters. No single point within it is special.

R code 14.41

```
post <- extract.samples(m14.8)

# plot the posterior median covariance function
plot( NULL , xlab="distance (thousand km)" , ylab="covariance" ,
    xlim=c(0,10) , ylim=c(0,2) )

# compute posterior mean covariance
x_seq <- seq( from=0 , to=10 , length.out=100 )
pmcov <- sapply( x_seq , function(x) post$etasq*exp(-post$rhosq*x^2) )
pmcov_mu <- apply( pmcov , 2 , mean )
lines( x_seq , pmcov_mu , lwd=2 )

# plot 60 functions sampled from posterior
for ( i in 1:50 )
    curve( post$etasq[i]*exp(-post$rhosq[i]*x^2) , add=TRUE ,
        col=col.alpha("black",0.3) )
```

FIGURE 14.11 shows the result. Each combination of values for ρ^2 and η^2 produces a relationship between covariance and distance. The posterior median function, shown by the thick curve, represents a center of plausibility. But the other curves show that there's a lot of uncertainty about the spatial covariance. Curves that peak at twice the posterior median peak, around 0.2, are commonplace. And curves that peak at half the median are very common,

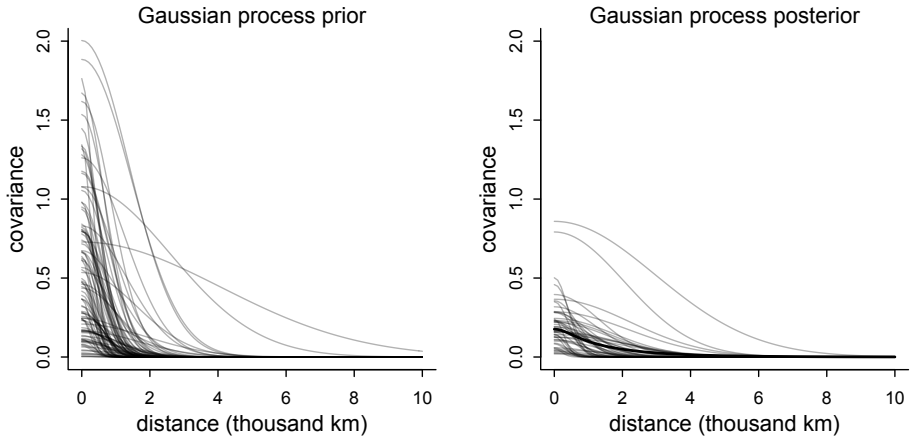


FIGURE 14.11. Let: Prior distribution of spatial covariance functions. Each curve shows a joint sample from the prior of ρ^2 and η^2. Right: Posterior distribution of the spatial covariance. The dark curve displays the posterior mean covariance at each distance. The thin curves show 50 functions sampled from the joint posterior distribution of ρ^2 and η^2.

as well. There's a lot of uncertainty about how strong the spatial effect is, but the majority of posterior curves decline to zero covariance before 4000 kilometers.

It's hard to interpret these covariances directly, because they are on the log-count scale, just like everything else in a Poisson GLM. So let's consider the correlations among societies that are implied by the posterior median. First, we push the parameters back through the function for **K**, the covariance matrix:

R code 14.42

```
# compute posterior median covariance among societies
K <- matrix(0,nrow=10,ncol=10)
for ( i in 1:10 )
    for ( j in 1:10 )
        K[i,j] <- median(post$etasq) *
                  exp( -median(post$rhosq) * islandsDistMatrix[i,j]^2 )
diag(K) <- median(post$etasq) + 0.01
```

Second, we convert `K` to a correlation matrix:

R code 14.43

```
# convert to correlation matrix
Rho <- round( cov2cor(K) , 2 )
# add row/col names for convenience
colnames(Rho) <- c("Ml","Ti","SC","Ya","Fi","Tr","Ch","Mn","To","Ha")
rownames(Rho) <- colnames(Rho)
Rho
```

```
     Ml   Ti   SC   Ya   Fi   Tr   Ch   Mn   To Ha
Ml 1.00 0.79 0.70 0.00 0.31 0.05 0.00 0.00 0.08  0
```

```
Ti 0.79 1.00 0.87 0.00 0.31 0.05 0.00 0.01 0.06  0
SC 0.70 0.87 1.00 0.00 0.17 0.11 0.01 0.02 0.02  0
Ya 0.00 0.00 0.00 1.00 0.00 0.01 0.16 0.14 0.00  0
Fi 0.31 0.31 0.17 0.00 1.00 0.00 0.00 0.00 0.61  0
Tr 0.05 0.05 0.11 0.01 0.00 1.00 0.09 0.56 0.00  0
Ch 0.00 0.00 0.01 0.16 0.00 0.09 1.00 0.32 0.00  0
Mn 0.00 0.01 0.02 0.14 0.00 0.56 0.32 1.00 0.00  0
To 0.08 0.06 0.02 0.00 0.61 0.00 0.00 0.00 1.00  0
Ha 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00  1
```

The cluster of small societies in the upper-left of the matrix—Malekula (Ml), Tikopia (Ti), and Santa Cruz (SC)—are highly correlated, all above 0.8 with one another. As you'll see in a moment, these societies are very close together, and they also have similar tool totals. These correlations were estimating with log population in the model, remember, and so suggest some additional resemblance even accounting for the average association between population and tools. On the other end of spectrum is Hawaii (Ha), which is so far from all of the other societies that the correlation decays to zero everyplace. Other societies display a range of correlations.

To make some sense of the variation in these correlations, let's plot them on a crude map of the Pacific Ocean. The `Kline2` data frame provides latitude and longitude for each society, to make this easy. I'll also scale the size of each society on the map in proportion to its log population.

R code 14.44

```
# scale point size to logpop
psize <- d$logpop / max(d$logpop)
psize <- exp(psize*1.5)-2

# plot raw data and labels
plot( d$lon2 , d$lat , xlab="longitude" , ylab="latitude" ,
    col=rangi2 , cex=psize , pch=16 , xlim=c(-50,30) )
labels <- as.character(d$culture)
text( d$lon2 , d$lat , labels=labels , cex=0.7 , pos=c(2,4,3,3,4,1,3,2,4,2) )

# overlay lines shaded by Rho
for( i in 1:10 )
    for ( j in 1:10 )
        if ( i < j )
            lines( c( d$lon2[i],d$lon2[j] ) , c( d$lat[i],d$lat[j] ) ,
                lwd=2 , col=col.alpha("black",Rho[i,j]^2) )
```

The result appears on the left side of FIGURE 14.12. Darker lines indicate stronger correlations, with pure white being zero correlation and pure black 100% correlation. The cluster of three close societies—Malekula, Tikopia, and Santa Cruz—stand out. Close societies have stronger correlations. But since we can't see total tools on this map, it's hard to see what the consequence of these correlations is supposed to be.

More sense can be made of these correlations, if we also compare against the simultaneous relationship between tools and log population. Here's a plot that combines the average posterior predictive relationship between log population and total tools with the shaded correlation lines for each pair of societies:

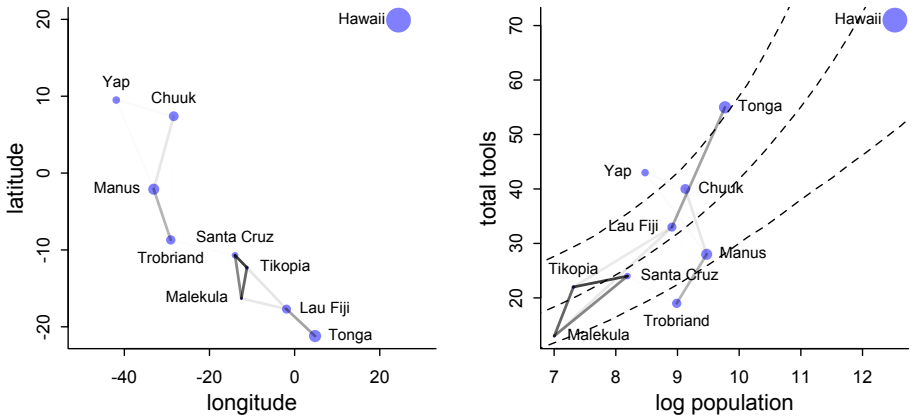


FIGURE 14.12. Left: Posterior correlations among societies in geographic space. Right: Same posterior correlations, now shown against relationship between total tools and log population.

R code 14.45

```
# compute posterior median relationship, ignoring distance
logpop.seq <- seq( from=6 , to=14 , length.out=30 )
lambda <- sapply( logpop.seq , function(lp) exp( post$a + post$bp*lp ) )
lambda.median <- apply( lambda , 2 , median )
lambda.PI80 <- apply( lambda , 2 , PI , prob=0.8 )

# plot raw data and labels
plot( d$logpop , d$total_tools , col=rangi2 , cex=psize , pch=16 ,
    xlab="log population" , ylab="total tools" )
text( d$logpop , d$total_tools , labels=labels , cex=0.7 ,
    pos=c(4,3,4,2,2,1,4,4,4,2) )

# display posterior predictions
lines( logpop.seq , lambda.median , lty=2 )
lines( logpop.seq , lambda.PI80[1,] , lty=2 )
lines( logpop.seq , lambda.PI80[2,] , lty=2 )

# overlay correlations
for( i in 1:10 )
    for ( j in 1:10 )
        if ( i < j )
            lines( c( d$logpop[i],d$logpop[j] ) ,
                   c( d$total_tools[i],d$total_tools[j] ) ,
                   lwd=2 , col=col.alpha("black",Rho[i,j]^2) )
```

This plot appears in the right-hand side of FIGURE 14.12. Now it's easier to appreciate that the correlations among Malekula, Tikopia, and Santa Cruz describe the fact that they are below the expected number of tools for their populations. All three societies lying below

the expectation, and being so close, is consistent with spatial covariance. The posterior correlations merely describe this feature of the data. Similarly, Manus and the Trobriands are geographically close, have a substantial posterior correlation, and fewer tools than expected for their population sizes. Tonga has more tools than expected for its population, and its proximity to Fiji counteracts some of the tug Fiji's smaller neighbors—Malekula, Tikopia, and Santa Cruz—exert on it. So the model seems to think Fiji would have fewer tools, if it weren't for Tonga.

Of course the correlations that this model describes by geographic distance may be the result of other, unmeasured commonalities between geographically close societies. For example, Manus and the Trobriands are geologically and ecologically quite different from Fiji and Tonga. So it could be availability of, for example, tool stone that explains some of the correlations. The Gaussian process regression is a grand and powerful descriptive model. As a result, its output is always compatible with many different causal explanations.

Rethinking: Dispersion by other names. The model in this section uses a Poisson likelihood, which is often sensitive to outliers, like the Hawaii data. You could use a gamma-Poisson likelihood instead, as explained in Chapter 12. But note that the varying effects in this example already induce additional dispersion around the Poisson mean. Adding Gaussian noise to each Poisson observation is another traditional way to handle over-dispersion in Poisson models. But do try the model with gamma-Poisson as well, so you can compare.

Overthinking: Non-centered islands. To build a non-centered Gaussian Process, we can use the same general trick of converting the covariance matrix to a Cholesky factor and then multiplying that factor by the z-scores of each varying effect. The covariance matrix is defined the same way. We just end up with some intermediate steps. Here is the Oceanic societies Gaussian Process model in non-centered form:

R code 14.46

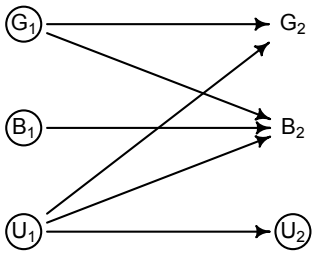
```
m14.8nc <- ulam(
    alist(
        T ~ dpois(lambda),
        lambda <- (a*P^b/g)*exp(k[society]),

        # non-centered Gaussian Process prior
        transpars> vector[10]: k <<- L_SIGMA * z,
        vector[10]: z ~ normal( 0 , 1 ),
        transpars> matrix[10,10]: L_SIGMA <<- cholesky_decompose( SIGMA ),
        transpars> matrix[10,10]: SIGMA <- cov_GPL2( Dmat , etasq , rhosq , 0.01 ),

        c(a,b,g) ~ dexp( 1 ),
        etasq ~ dexp( 2 ),
        rhosq ~ dexp( 0.5 )
    ), data=dat_list , chains=4 , cores=4 , iter=2000 )
```

The new element above is the Stan function `cholesky_decompose`, which takes covariance (or correlation) matrix and returns its Cholesky factor. That Cholesky factor can then be mixed with z-scores as before to produce varying effects on the right scale. If you check the posterior, you'll see this version samples more efficiently. As always, the cost is that the model is harder to read. With a very large `SIGMA` matrix, often there is no choice but to use the Cholesky (non-centered) parameterization. The next example, for example, is like this.

14.5.2. Example: Phylogenetic distance. Species, like islands, are more or less distance from one another. However their distance is not physical but rather temporal—how long since a common ancestor? Evolutionary biologists investigate how phylogenetic relationships influence patterns of variation in the bodies and brains of different species. It's a fact that species with more recent common ancestors have higher trait correlations. Do these correlations matter?

Phylogenetic distance can have two important causal influences. The first is that two species that only recently separated tend to be more similar, assuming their traits are not maintained by selection but rather drifting neutrally around. The second causal influence is indirect. Phylogenetic distance is a proxy for unobserved variables that generate covariation among species, even when selection matters. Closely related species likely share more of these, but distantly related species share many fewer. For example, all mammals nurse their young with milk. Flight in birds similarly influences many traits. These discrete, life history altering traits can have strong causal influence on other traits. When not observed, phylogenetic distance is a potentially useful proxy for these variables. But only if the trait model captures the right details.[206] These methods do not just work automatically, as they are too often ritually presented in journals.

Consider as an example the causal influence of group size (G) on brain size (B). Hypotheses connecting these variables are popular, because primates (including humans) are unusual in both. Most primates live in social groups. Most mammals do not. Second, primates have relatively large brains. There is a family of hypotheses linking these two features. Suppose for example that group living, whatever its cause, could select for larger brains, because once you live with others, a larger brain helps to cope with the complexity of cooperation and manipulation. This hypothesis implies a causal time series. Let's draw it:

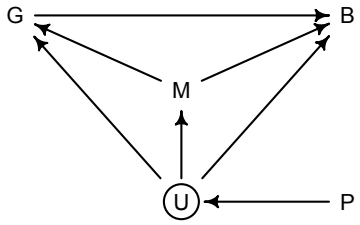


The subscripts are time points in the evolutionary history of different populations. So G_1 and group size at time 1 and G_2 is group size in the next time point. There are plausibly many potential confounds, shown here as U_1 and U_2. Each variable influences itself in the next time step, as you might expect in an evolving system. There is also a causal influences of G_1 on B_2—a species' recent group size influenced its current brain size. This is what we'd like to estimate. However the confounds U_1 also possibly influence everything. As in previous examples, circled variables are unobserved. So we can't just condition on U_1 to block confounding. We also don't even have G_1 to use in a model, but only its descendant G_2. But note that if we did have measurements of G_1 and U_1, we could use these and not worry at all about phylogeny.

Since we haven't observed the past, we need some way to estimate its influence. This is where the branching history of the species might help. Phylogeny is associated with the patterns of covariation across species, because recently diverged species tend to be more similar. So phylogenetic relationships, expressed as distance, can be used to partially reconstruct

confounds. This depends upon having both a good phylogeny and a good model of the relationship between phylogenetic distance and trait evolution. Neither is a trivial problem. But the approach is justified in theory, if not always possible in practice.

It will help to draw this approach and then use it in an actual model.



There's a lot going on here, but we can take it one piece at a time. Again, we're interested in $G \rightarrow B$. There is one confound we know for sure, body mass (M). It possibly influences both G and B. So we'll include that in the model. The unobserved confounds U could potentially influence all three variables. Finally, we let the phylogenetic relationships (P) influence U. How is P causal? If we traveled back in time and delayed a split between two species, it could influence the expected differences in their traits. So it is really the timing of the split that is causal, not the phylogeny. Of course P may also influence G and B and M directly. But those arrows aren't our concern right now, so I've omitted them for clarity.

We want to be sure any association between group size G and brain size B is not through a backdoor. As always, we look for all the paths between G and B, identify which are backdoors, and consider if there are any methods for closing the backdoor paths. In the DAG above, there are backdoor paths through M and through U. We can condition on M to block that confound. But we can't condition on U. But if we can use P to somehow reconstruct the covariation that U induces between G and B, that could be enough.

That's the strategy. Now implementing that strategy is famously hard. GLMs that try to include phylogenetic distance often go by the name **PHYLOGENETIC REGRESSION**. The original phylogenetic regression approach treats phylogenetic distance in a highly constrained and unrealistic way, based on a neutral model of divergence with time.[207] There are many variants. But all of them use some function of phylogenetic distance to model the covariation among species. So learning the basic phylogenetic regression model helps bootstrap your understanding, even though you really should use something better in your own analyses. After introducing the basic phylogenetic regression, I'll show you how to more flexibly model phylogenetic distance. There is no universally correct function that maps phylogeny onto the confounds that matter. So flexibility is needed.

To begin, load the primates data and its phylogeny as well:

R code 14.47

```
library(rethinking)
data(Primates301)
data(Primates301_nex)

# plot it using ape package - install.packages('ape') if needed
library(ape)
plot( ladderize(Primates301_nex) , type="fan" , font=1 , no.margin=TRUE ,
    label.offset=1 , cex=0.5 )
```

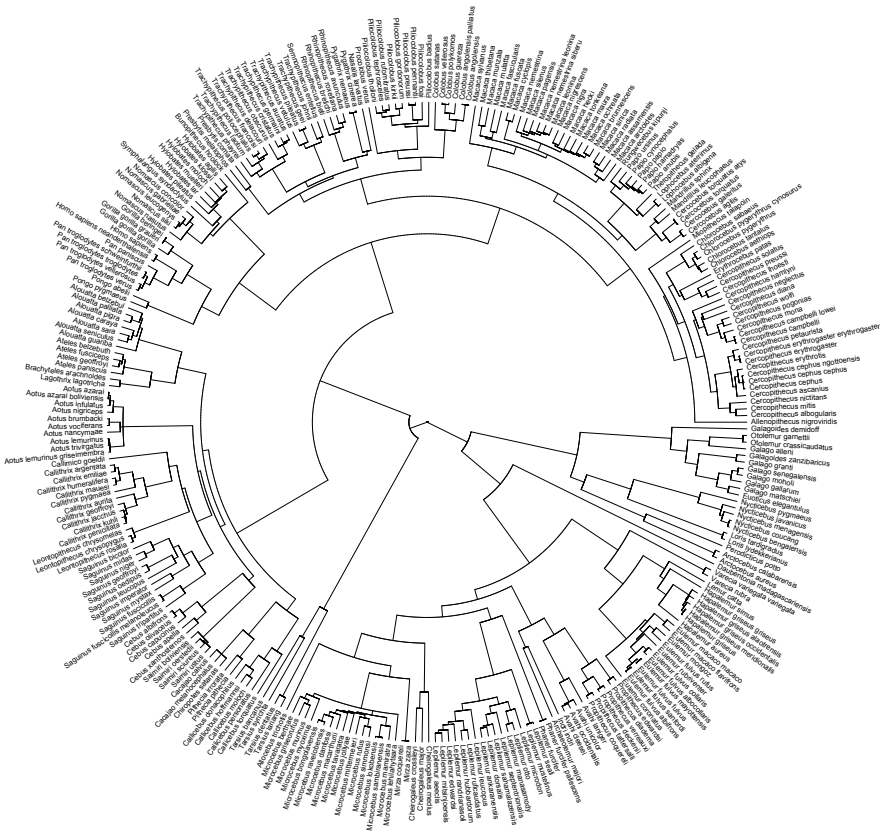


FIGURE 14.13. Consensus phylogeny for 301 primate species. See the citations in `?Primates301` for sources.

I've plotted this phylogeny as FIGURE 14.13. We're going to use this tree as a way to model unobserved confounds. At the same time, we'd like to deal with the fact that some groups of closely related species may be over-represented in nature. There are lots of lemurs for example. This produces an imbalance in sampling issue, analogous to an ordinary multilevel modeling context. And varying effects can help us here as well. But we'll get the varying effects, as it were, from the phylogenetic tree structure.

Before we do anything with the tree, however, let's run an ordinary regression analyzing (log) group size as a function of (log) brain size and (log) body size. But I want to build this ordinary regression in an un-ordinary style, because it will help you understand the next step, where we stick the phylogenetic information inside. Think of all of the species as a single variable, a vector of 301 trait values. Of course some of these values are more similar to one another. In a typical regression, we model those similarities using predictor variables. After conditioning on the predictor variables, the model expects correlations. So we can write such a model using a big, multi-variate outcome distribution. It looks like this:

$$\mathbf{B} \sim \text{MVNormal}(\mu, \mathbf{S})$$
$$\mu_i = \alpha + \beta_G G_i + \beta_M M_i$$

where **B** is a vector of species brain sizes and **S** is a covariance matrix with as many rows and columns as there are species. In an ordinary regression, this matrix takes the form:

$$\mathbf{S} = \sigma^2\mathbf{I}$$

where σ is the same standard deviation you've used since Chapter 4 and **I** is an **IDENTITY MATRIX**, which is just a matrix with 1 along the diagonal and zeros everywhere else. You can think of it as a correlation matrix in which all of the correlations are zero. So multiplying the variance into it just gives each species the same (residual) variance. It's an ordinary linear regression, but thought of as having a single, multi-variate outcome.

Let's fit this model to the primate data. First we need to trim down to the species for which we have group size, brain size, and body size data:

R code 14.48

```
d <- Primates301
d$name <- as.character(d$name)
dstan <- d[ complete.cases( d$group_size , d$body , d$brain ) , ]
spp_obs <- dstan$name
```

You should have 151 species left. Now to make a list with standardized logged variables and pass it all to `ulam`:

R code 14.49

```
dat_list <- list(
    N_spp = nrow(dstan),
    M = standardize(log(dstan$body)),
    B = standardize(log(dstan$brain)),
    G = standardize(log(dstan$group_size)),
    Imat = diag(nrow(dstan)) )

m14.9 <- ulam(
    alist(
        B ~ multi_normal( mu , SIGMA ),
        mu <- a + bM*M + bG*G,
        matrix[N_spp,N_spp]: SIGMA <- Imat * sigma_sq,
        a ~ normal( 0 , 1 ),
        c(bM,bG) ~ normal( 0 , 0.5 ),
        sigma_sq ~ exponential( 1 )
    ), data=dat_list , chains=4 , cores=4 )
precis( m14.9 )
```

```
          mean   sd  5.5% 94.5% n_eff Rhat
a         0.00 0.02 -0.03  0.03  1859    1
bG        0.12 0.02  0.09  0.16  1572    1
bM        0.89 0.02  0.86  0.93  1481    1
sigma_sq 0.05 0.01  0.04  0.06  2040    1
```

Looks like a reliably positive association between brain size and group size, as well as a strong association between body mass and brain size. There is no basis yet to interpret these associations causally, because we know these data are swirling with confounds.

Now we'll conduct two different kinds of phylogenetic regression. In both, all we have to do is replace the covariance matrix **S** above with a different matrix that encodes some phylogenetic information. The first regression is one of the oldest and most conservative,

a **Brownian motion** interpretation of the phylogeny that implies a very particular covariance matrix. Brownian motion just means Gaussian random walks. If species traits drift randomly with respect to one another after speciation, then the covariance between a pair of species ends up being linearly related to the phylogenetic branch distance between them—the further apart, the less covariance, as a proportion of distance. Of course the traits we are interested in obviously do not evolve neutrally, and they also evolve at different rates in different parts of the tree. But what you are about to do is unfortunately the most common formethodm of phylogenetic control.

Let's compute the implied covariance matrix, the distance matrix, and show how they are related. The `ape` R package has all of the functions you need.

R code 14.50

```
library(ape)
tree_trimmed <- keep.tip( Primates301_nex, spp_obs )
Rbm <- corBrownian( phy=tree_trimmed )
V <- vcv(Rbm)
Dmat <- cophenetic( tree_trimmed )
plot( Dmat , V , xlab="phylogenetic distance" , ylab="covariance" )
```

I don't display the plot here, but if you run the above code, you'll see a scatterplot with pairs of species as points. The horizontal axis is phylogenetic, or patristic, distance. The vertical is the covariance under the Brownian model. They are really just inverses of one another. You can see this even more clearly if you use `image(V)` and `image(Dmat)` to plot heat maps of each.

Now we can just insert this new matrix into our regression. The is otherwise the same. But first we need to get the rows and columns in the same order as the rest of the data and then convert it to a correlation matrix, so we can estimate the residual variance. Then we can just replace the identity matrix with our new correlation matrix and go.

R code 14.51

```
# put species in right order
dat_list$V <- V[ spp_obs , spp_obs ]
# convert to correlation matrix
dat_list$R <- dat_list$V / max(V)

# Brownian motion model
m14.10 <- ulam(
    alist(
        B ~ multi_normal( mu , SIGMA ),
        mu <- a + bM*M + bG*G,
        matrix[N_spp,N_spp]: SIGMA <- R * sigma_sq,
        a ~ normal( 0 , 1 ),
        c(bM,bG) ~ normal( 0 , 0.5 ),
        sigma_sq ~ exponential( 1 )
    ), data=dat_list , chains=4 , cores=4 )
precis( m14.10 )
```

```
          mean   sd  5.5% 94.5% n_eff Rhat
a        -0.20 0.17 -0.47  0.06  2152    1
bG       -0.01 0.02 -0.04  0.02  2691    1
bM        0.70 0.04  0.64  0.76  1935    1
```

```
sigma_sq  0.16 0.02  0.13  0.19  2251    1
```

This model annihilates group size—the posterior mean is almost zero and there is a lot of mass on both sides of zero. The big change from the previous model suggests that there is a lot of clustering of brain size in the tree and that this produces a spurious relationship with group size, which also clusters in the tree. How the model uses this clustering depends upon the details of the correlation matrix we gave it.

The Brownian motion model is a special kind of Gaussian process in which the covariance declines in a very rigid way with increasing distance. There is no need to be so rigid and good reason to think evolution is not well-described by Brownian motion. It's very common to use something called **Pagel's lambda** to modify the Brownian motion model. But all this does is scale all of the species correlations by a common factor. It maintains the same arbitrary and unrealistic distance model. Another common alternative is the **Ornstein–Uhlenbeck process** (or OU process), which is a damped Brownian motion process that tends to return towards some mean (or means). What this does in practice is constrain the variation, making the relationship between phylogenetic distance and covariance nonlinear.[208] More precisely, the OU process just defines the covariance between two species i and j as:

$$K(i,j) = \eta^2 \exp(-\rho^2 D_{ij})$$

This is an exponential distance kernel, unlike the quadratic kernel in the previous example. The exponential kernel says that covariance between points (species) declines rapidly, making for much less smooth functions. It is also usually harder to fit to data, since it a much rougher function. This means in practice that you'll need to be careful about priors, potentially making them narrower.

But the OU process is still a Gaussian process, and you can fit it the same way as the quadratic kernel in the previous section. The literature on phylogenetic regression has not emphasized this fact. But expressing the model as a Gaussian process makes it possible to customize the function space as the problem requires.[209] This framing isn't yet common. Biologists tend to use phylogenies under a cloud of superstition and fearful button pushing. But the Gaussian process framing both unifies existing approaches and allows the model to describe the pattern of covariation with phylogenetic distance. Hopefully it also makes clear that there is no correct way to include phylogenetic distance. If the goal is estimate a causal effect, then it isn't good enough to reject some null model. We need to usefully reconstruct patterns among unmeasured confounds. And different evolutionary histories will require different models.

To build the Gaussian process regression, we need a distance matrix. We already have that—you computed it earlier. Then we just need the Gaussian process construction line of code. In this example, we'll use the OU process kernel, which is known more generally as the L1 norm, which `ulam` provides as `cov_GPL1`. But see the Overthinking box further down, to see how to write your own Gaussian process kernels.

R code 14.52

```
# add scaled and reordered distance matrix
dat_list$Dmat <- Dmat[ spp_obs , spp_obs ] / max(Dmat)

m14.11 <- ulam(
    alist(
        B ~ multi_normal( mu , SIGMA ),
```

```
        mu <- a + bM*M + bG*G,
        matrix[N_spp,N_spp]: SIGMA <- cov_GPL1( Dmat , etasq , rhosq , 0.01 ),
        a ~ normal(0,1),
        c(bM,bG) ~ normal(0,0.5),
        etasq ~ half_normal(1,0.25),
        rhosq ~ half_normal(3,0.25)
    ), data=dat_list , chains=4 , cores=4 )
precis( m14.11 )
```

```
       mean   sd  5.5% 94.5% n_eff Rhat
a     -0.07 0.08 -0.19  0.06  2168    1
bG     0.05 0.02  0.01  0.09  2634    1
bM     0.83 0.03  0.79  0.88  2280    1
etasq  0.03 0.01  0.03  0.05  2060    1
rhosq  2.79 0.26  2.36  3.20  2192    1
```

Now group size is seemingly associated with brain size again. The association is small, but most of the posterior mass is above zero. Why are the results different? The answer must be that the inferred covariance function looks rather different than the Brownian motion model. So let's look at the posterior covariance functions implied by `etasq` and `rhosq`. Remember that these two parameters interact to produce the covariance function, and they are almost always strongly correlated in the posterior, so you can't really see what's going on by looking at them separately. We need to extract them and push them back through the Gaussian process covariance function:

R code 14.53

```
post <- extract.samples(m14.11)
plot( NULL , xlim=c(0,max(dat_list$Dmat)) , ylim=c(0,1.5) ,
    xlab="phylogenetic distance" , ylab="covariance" )

# posterior
for ( i in 1:30 )
    curve( post$etasq[i]*exp(-post$rhosq[i]*x) , add=TRUE , col=rangi2 )

# prior mean and 89% interval
eta <- abs(rnorm(1e3,1,0.25))
rho <- abs(rnorm(1e3,3,0.25))
d_seq <- seq(from=0,to=1,length.out=50)
K <- sapply( d_seq , function(x) eta*exp(-rho*x) )
lines( d_seq , colMeans(K) , lwd=2 )
shade( apply(K,2,PI) , d_seq )
text( 0.5 , 0.5 , "prior" )
text( 0.2 , 0.1 , "posterior" , col=rangi2 )
```

The result is show in FIGURE 14.14. The horizontal axis is the standardized phylogenetic distance—1 just means the longest distance in the sample. The vertical axis is covariance. The blue curves are 30 draws from the posterior distribution. The black curve is the prior mean. The posterior is pressed up against the bottom axis, indicating a very low covariance between species at any distance. There just isn't a lot of phylogenetic covariance for brain sizes, at least according to this model and these data. As a result, the phylogenetic distance

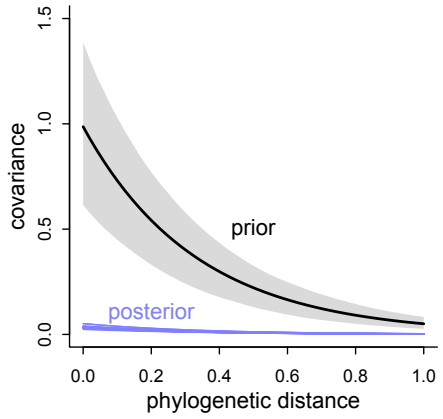


FIGURE 14.14. Posterior covariance functions for the Gaussian process phylogenetic regression (blue), compared to the prior (gray). Unlike the Brownian motion model, in which covariance starts high and decays linearly with distance, this model favors a very small covariation at all distances.

doesn't completely explain away the association between group size and brain size, as it did in the Brownian motion model.

Overthinking: Building custom kernels. The `rethinking` package provides `cov_GPL1` (the OU kernel) and `cov_GPL2` (the quadratic kernel) for building Gaussian process covariance matrices. But it's easy to build your own, if you use Stan directly. Let's look at `stancode(m14.11)`. The top part is a custom `functions` block, containing the `cov_GPL1` function:

```
functions{
    matrix cov_GPL1(matrix x, real sq_alpha, real sq_rho, real delta) {
        int N = dims(x)[1];
        matrix[N, N] K;
        for (i in 1:(N-1)) {
          K[i, i] = sq_alpha + delta;
          for (j in (i + 1):N) {
            K[i, j] = sq_alpha * exp(-sq_rho * x[i,j] );
            K[j, i] = K[i, j];
          }
        }
        K[N, N] = sq_alpha + delta;
        return K;
    }
}
```

This function takes as input as distance matrix `x` and the parameters of the Gaussian process. It then loops over all the cells in the covariance matrix `K`, computing the value of each. To modify the kernel, you'd change the line that computes each covariance:

```
K[i, j] = sq_alpha * exp(-sq_rho * x[i,j] );
```

For example, the quadratic kernel just squares the `x[i,j]`. All that remain is to call the function inside the `model` block.

14.6. Summary

This chapter extended the basic multilevel strategy of partial pooling to slopes as well as intercepts. Accomplishing this meant modeling covariation in the statistical population

of parameters. The LKJcorr prior was introduced as a convenient family of priors for correlation matrices. You saw how covariance models can be applied to causal inference, using instrumental variables and the front-door criterion. Gaussian processes represent a practical method of extending the varying effects strategy to continuous dimensions of similarity, such as spatial, network, phylogenetic, or any other abstract distance between entities in the data. The next chapter continues to develop the broader multilevel approach by applying it to commonplace problems in statistical inference: measurement error and missing data.

14.7. Practice

Easy.

14E1. Add to the following model varying slopes on the predictor x.

$$\begin{aligned} y_i &\sim \text{Normal}(\mu_i, \sigma) \\ \mu_i &= \alpha_{\text{GROUP}[i]} + \beta x_i \\ \alpha_{\text{GROUP}} &\sim \text{Normal}(\alpha, \sigma_\alpha) \\ \alpha &\sim \text{Normal}(0, 10) \\ \beta &\sim \text{Normal}(0, 1) \\ \sigma &\sim \text{HalfCauchy}(0, 2) \\ \sigma_\alpha &\sim \text{HalfCauchy}(0, 2) \end{aligned}$$

14E2. Think up a context in which varying intercepts will be positively correlated with varying slopes. Provide a mechanistic explanation for the correlation.

14E3. When is it possible for a varying slopes model to have fewer effective parameters (as estimated by WAIC or DIC) than the corresponding model with fixed (unpooled) slopes? Explain.

Medium.

14M1. Repeat the café robot simulation from the beginning of the chapter. This time, set `rho` to zero, so that there is no correlation between intercepts and slopes. How does the posterior distribution of the correlation reflect this change in the underlying simulation?

14M2. Fit this multilevel model to the simulated café data:

$$\begin{aligned} W_i &\sim \text{Normal}(\mu_i, \sigma) \\ \mu_i &= \alpha_{\text{CAFÉ}[i]} + \beta_{\text{CAFÉ}[i]} A_i \\ \alpha_{\text{CAFÉ}} &\sim \text{Normal}(\alpha, \sigma_\alpha) \\ \beta_{\text{CAFÉ}} &\sim \text{Normal}(\beta, \sigma_\beta) \\ \alpha &\sim \text{Normal}(0, 10) \\ \beta &\sim \text{Normal}(0, 10) \\ \sigma &\sim \text{HalfCauchy}(0, 1) \\ \sigma_\alpha &\sim \text{HalfCauchy}(0, 1) \\ \sigma_\beta &\sim \text{HalfCauchy}(0, 1) \end{aligned}$$

Use WAIC to compare this model to the model from the chapter, the one that uses a multi-variate Gaussian prior. Explain the result.

14M3. Re-estimate the varying slopes model for the `UCBadmit` data, now using a non-centered parameterization. Compare the efficiency of the forms of the model, using `n_eff`. Which is better? Which chain sampled faster?

14M4. Use WAIC to compare the Gaussian process model of Oceanic tools to the models fit to the same data in Chapter 11. Pay special attention to the effective numbers of parameters, as estimated by WAIC.

14M5. Modify the phylogenetic distance example to use group size as the outcome and brain size as a predictor. Assuming brain size influences group size, what is your estimate of the effect? How does phylogeny influence the estimate?

Hard.

14H1. Let's revisit the Bangladesh fertility data, `data(bangladesh)`, from the practice problems for Chapter 13. Fit a model with both varying intercepts by `district_id` and varying slopes of `urban` by `district_id`. You are still predicting `use.contraception`. Inspect the correlation between the intercepts and slopes. Can you interpret this correlation, in terms of what it tells you about the pattern of contraceptive use in the sample? It might help to plot the mean (or median) varying effect estimates for both the intercepts and slopes, by district. Then you can visualize the correlation and maybe more easily think through what it means to have a particular correlation. Plotting predicted proportion of women using contraception, with urban women on one axis and rural on the other, might also help.

14H2. Varying effects models are useful for modeling time series, as well as spatial clustering. In a time series, the observations cluster by entities that have continuity through time, such as individuals. Since observations within individuals are likely highly correlated, the multilevel structure can help quite a lot. You'll use the data in `data(Oxboys)`, which is 234 height measurements on 26 boys from an Oxford Boys Club (I think these were like youth athletic leagues?), at 9 different ages (centered and standardized) per boy. You'll be interested in predicting `height`, using `age`, clustered by `Subject` (individual boy).

Fit a model with varying intercepts and slopes (on age), clustered by Subject. Present and interpret the parameter estimates. Which varying effect contributes more variation to the heights, the intercept or the slope?

14H3. Now consider the correlation between the varying intercepts and slopes. Can you explain its value? How would this estimated correlation influence your predictions about a new sample of boys?

14H4. Use `mvrnorm` (in `library(MASS)`) or `rmvnorm` (in `library(mvtnorm)`) to simulate a new sample of boys, based upon the posterior mean values of the parameters. That is, try to simulate varying intercepts and slopes, using the relevant parameter estimates, and then plot the predicted trends of height on age, one trend for each simulated boy you produce. A sample of 10 simulated boys is plenty, to illustrate the lesson. You can ignore uncertainty in the posterior, just to make the problem a little easier. But if you want to include the uncertainty about the parameters, go for it.

Note that you can construct an arbitrary variance-covariance matrix to pass to either `mvrnorm` or `rmvnorm` with something like:

R code 14.54
```
S <- matrix( c( sa^2 , sa*sb*rho , sa*sb*rho , sb^2 ) , nrow=2 )
```

where `sa` is the standard deviation of the first variable, `sb` is the standard deviation of the second variable, and `rho` is the correlation between them.