STATISTICAL RETHINKING 2022
WEEK 7 SOLUTIONS

1. The necessary model has the same structure as the tadpoles model.

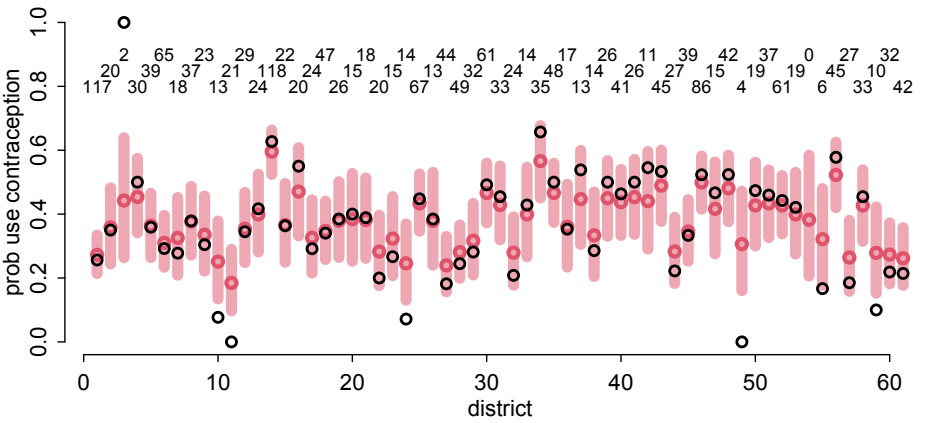
```
data(bangladesh)
d <- bangladesh
dat <- list(
    C = d$use.contraception,
    D = as.integer(d$district) )
m1 <- ulam(
    alist(
        C ~ bernoulli(p),
        logit(p) <- a[D],
        vector[61]:a ~ normal(abar,sigma),
        abar ~ normal(0,1),
        sigma ~ exponential(1)
    ) , data=dat , chains=4 , cores=4 )
```

We could look at the precis output, but that isn't so useful. Let's go ahead and plot the individual district estimates:

```
# plot estimates
post <- extract.samples(m1)
p <- inv_logit(post$a)
plot( apply(p,2,mean) , xlab="district" , lwd=3 , col=2 ,
    ylim=c(0,1) , ylab="prob use contraception" )
for ( i in 1:61 ) lines( c(i,i) , PI(p[,i]) , lwd=8 ,
    col=col.alpha(2,0.5) )

# show raw proportions - have to skip 54
n <- table(dat$D)
Cn <- xtabs(dat$C ~ dat$D)
pC <- as.numeric( Cn/n )
pC <- c( pC[1:53] , NA , pC[54:60] )
points( pC , lwd=2 )

# add sample size labels
y <- rep(c(0.8,0.85,0.9),len=61)
n <- as.numeric(n)
n <- c( n[1:53] , 0 , n[54:60] )
text( 1:61 , y , n , cex=0.8 )
```

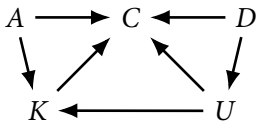


The black points are the raw estimates, just the portion of women using contraception in the sample in each district. The red points are the partial pooling estimates. The pink bands are the compatibility intervals. The numbers are the sample sizes in each district.

We can see some districts in which the black point is entirely outside the pink region. These districts tend to have low sample sizes. Consider the third district from the left. It had only 2 women in the sample, and both used contraception. The model is quite skeptical that the district has 100% contraceptive use. There are similar but less dramatic pooling effects in other districts. Note however in each case that the estimate band gets pulled in the direction of the black point as well.

District 54 is the one with zero women sampled. You can find it on the right side of the plot, with no black point. The estimate there is the population expectation, informed by the other districts.

2. Many sensible DAGs are possible, but there will all be quite similar. Here is my DAG:



The key features of this model are that U influences K and that D influences U.

For the total causal effect of U on C, we have to close the backdoor path through D. That is it. The path through K is part of the causal effect of U. As a competing cause, we could also stratify by A to increase precision. But it isn't necessary for identification.

For the direct effect of U on C, we have to also stratify by K. But when we do that, it open a collider path through A. So we must stratify by D, K, and A to estimate the direct effect of U.

3. The total effect just needs us to add U to the model from problem 1:

```
dat <- list(
    C = d$use.contraception,
    D = as.integer(d$district),
    U = d$urban,
    A = standardize(d$age.centered),
    K = d$living.children )

# total
m3a <- ulam(
    alist(
        C ~ bernoulli(p),
        logit(p) <- a[D] + bU*U,
        bU ~ normal(0,0.5),
        vector[61]:a ~ normal(abar,sigma),
        abar ~ normal(0,1),
        sigma ~ exponential(1)
    ) , data=dat , chains=4 , cores=4 )
precis(m3a)
```

```
       mean   sd  5.5% 94.5% n_eff Rhat4
bU     0.61 0.11  0.43  0.78  1181  1.00
abar  -0.68 0.09 -0.82 -0.55   854  1.01
sigma  0.46 0.08  0.34  0.59   454  1.01
```

So a positive total effect, corresponding to about 80% higher odds on average for women in urban parts of districts.

Now the model for the direct effect must include all of the variables. But how we include them is up to us. I will use an ordered monotomic effect for K. There are only the values 1 to 4 observed. So we only need 4 parameters. It it tempting to include A in the same way, but we only have it as a centered value, so we don't know the actual ages (although it isn't too hard to reconstruct them from the relative values). Here's my model:

```
# direct
dat$Kprior <- rep(2,3)
m3b <- ulam(
    alist(
        C ~ bernoulli(p),
        logit(p) <- a[D] + bU*U + bA*A +
                    bK*sum( delta_j[1:K] ),
        c(bU,bA,bK) ~ normal(0,0.5),
        vector[4]: delta_j <<- append_row( 0 , delta ),
        simplex[3]: delta ~ dirichlet( Kprior ),
        vector[61]:a ~ normal(abar,sigma),
        abar ~ normal(0,1),
        sigma ~ exponential(1)
```

```
) , data=dat , chains=4 , cores=4 , iter=4000 )
precis(m3b,depth=2)
```

```
          mean   sd  5.5% 94.5% n_eff Rhat4
bK        1.25 0.15  1.01  1.49   918     1
bA       -0.23 0.06 -0.33 -0.13  2556     1
bU        0.68 0.12  0.49  0.87  3720     1
delta[1]  0.74 0.08  0.60  0.86  6911     1
delta[2]  0.17 0.08  0.05  0.31  7075     1
delta[3]  0.09 0.05  0.02  0.19  9288     1
a[1]     -2.26 0.24 -2.65 -1.88  1880     1
...
a[61]    -2.10 0.32 -2.62 -1.60  2855     1
abar     -1.55 0.14 -1.77 -1.32   848     1
sigma     0.48 0.08  0.36  0.62  2201     1
```

I abbreviated the intercept output, but you get the idea. The direct effect of U, according to this model, is basically unchanged. This implies that there is no strong effect through K.

I didn't ask you to interpret the other coefficients. The K estimates are interesting through. According to the DAG, the estimates for K should estimate its direct effect on C. The `delta` parameters are the incremental effects of each child (after the first). So the largest effect is the second child, which accounts for 74% (60%-86%) of the effect, which is about $1.25 \times 0.74 = 0.925$ on the log-odds scale, which corresponds to a more than doubling of the odds of contraception.

In the models above, I did not stratify the effect of U by D. But it is reasonable also to let `bU` vary by district. So let's do that, on the notion that all of the effects on C could be moderated by unobserved features of each district.

Here's the revised total effect model, now with a covariance matrix for the varying effects. I use a single vector `a` for all of the district parameters.

```
# total
m3c <- ulam(
    alist(
        C ~ bernoulli(p),
        logit(p) <- a[D,1] + a[D,2]*U,
        matrix[61,2]:a ~ multi_normal(abar,Rho,sigma),
        vector[2]:abar ~ normal(0,1),
        corr_matrix[2]:Rho ~ lkj_corr(4),
        vector[2]:sigma ~ exponential(1)
) , data=dat , chains=4 , cores=4 , iter=4000 )
precis(m3c,depth=2)
```

```
          mean   sd  5.5% 94.5% n_eff Rhat4
abar[1]  -0.70 0.10 -0.86 -0.55  4950     1
```

```
abar[2]   0.68 0.17  0.41  0.94  3354     1
sigma[1]  0.56 0.09  0.42  0.72  1794     1
sigma[2]  0.76 0.20  0.46  1.08   974     1
```

The average effects are basically unchanged from the previous model, but there is a lot of variation across districts in the effect of U. You can see that from the `sigma[2]` estimate.

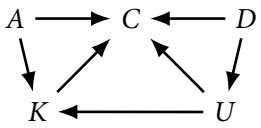
Now for the direct effect. This model really needs the non-centered parameterization to sample efficiently.

```
# total
m3dnc <- ulam(
    alist(
        C ~ bernoulli(p),
        logit(p) <- abar[1] + a[D,1] +
                    (abar[2] + a[D,2])*U +
                    (abar[3] + a[D,3])*A +
                    (abar[4] + a[D,4])*sum( delta_j[1:K] ),
        transpars> matrix[61,4]:a <-
            compose_noncentered( sigma , L_Rho , Z ),
        matrix[4,61]:Z ~ normal( 0 , 1 ),
        vector[4]: delta_j <<- append_row( 0 , delta ),
        simplex[3]: delta ~ dirichlet( Kprior ),
        vector[4]:abar ~ normal(0,1),
        cholesky_factor_corr[4]:L_Rho ~ lkj_corr_cholesky( 4 ),
        vector[4]:sigma ~ exponential(1),
        gq> matrix[4,4]:Rho <<- Chol_to_Corr( L_Rho )
    ) , data=dat , chains=4 , cores=4 , iter=4000 )
precis(m3dnc,depth=3,pars=c("abar","sigma"))
```

```
          mean   sd  5.5% 94.5% n_eff Rhat4
abar[1]  -1.65 0.16 -1.90 -1.40  4231  1.00
abar[2]   0.74 0.17  0.48  1.01  5116  1.00
abar[3]  -0.26 0.07 -0.37 -0.15  6900  1.00
abar[4]   1.37 0.18  1.09  1.66  5496  1.00
sigma[1]  0.60 0.13  0.40  0.83  2207  1.00
sigma[2]  0.71 0.21  0.38  1.06  2254  1.00
sigma[3]  0.11 0.08  0.01  0.26  2613  1.00
sigma[4]  0.33 0.21  0.03  0.69   709  1.01
```

The average effect for U is a little larger, but not much. There isn't as much variation in the effect of K and A, the [3] and [4] elements of `sigma`.

4-OPTIONAL CHALLENGE. This is a very hard problem, so if you made any progress at all, you are doing great. This kind of task is complicated, because when we imagine an intervention on one variable, it can put the entire DAG into play. To make this clear, here's out DAG again:



When we change U in any particular district, the effect depends upon all of the arrows coming out of U. So any effect through K as well. We also have to deal with moderators, because this is a non-linear model. So the influence of A matters, as well as the precise distribution of A that we assume. And since I assumed A acts partly through K, we need to include the effect of A on K. That means we need to model both the influences on C and on K. So two models to estimate.

Finally, if we are trying to estimate a real intervention (as opposed to a purely hypothetical one), then the distribution of U in each district before intervention will matter, since that will adjust the total possible impact of changing U. I will focus on a hypothetical, going from no urban living to all urban living, to make the example simpler.

So what do we need to get started here? We need a model to measure the total effect of U and the influence of A, all stratified by D. Here's a model that takes the full model from the previous problem and inserts another model for K. I'm going to simplify this a bit by making K a standardized variable and doing a simple linear model with it. That is just for transparency. Ideally, we would use an ordered logit or such for K.

```
dat$Ks <- standardize(dat$K)
m4 <- ulam(
    alist(
        # C model
        C ~ bernoulli(p),
        logit(p) <- abar[1] + a[D,1] +
                    (abar[2] + a[D,2])*U +
                    (abar[3] + a[D,3])*A +
                    (abar[4] + a[D,4])*Ks,

        # K model
        Ks ~ normal( mu , tau ),
        mu <- aK + bAK*A + bUK*U,
        c(aK,bAK,bUK) ~ normal(0,1),
        tau ~ exponential(1),

        # guts of the machine below
        transpars> matrix[61,4]:a <-
            compose_noncentered( sigma , L_Rho , Z ),
        matrix[4,61]:Z ~ normal( 0 , 1 ),
        vector[4]:abar ~ normal(0,1),
        cholesky_factor_corr[4]:L_Rho ~ lkj_corr_cholesky( 4 ),
        vector[4]:sigma ~ exponential(1)
```

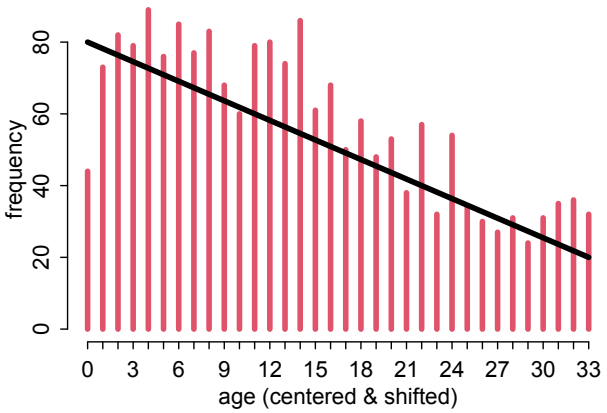
```
    ) , data=dat , chains=4 , cores=4 )
```

Now it is just a matter of (1) assuming some age distribution, (2) picking a district, and (3) simulating outcomes. Step 3 can be complicated. I made some simplifications to the above model, so that we can use the automated sim() method in the rethinking package. But for a more elaborate model, we could always do a custom simulation. The principles are the same. Conceptually it isn't so bad: (1) simulate K using U and A, (2) simulate C using A and U and K and D. Do that for a representative sample of the ages, for each treatment value of U you like, and then compare across U values. The code can be messy though.

Let's start by adopting a standard age distribution. The distribution in the total sample is reasonable, but obviously under-samples in some places and over-samples in others. So I'll put a linear trend on it:

```
# age distribution - total sample
Ax <- round( d$age.centered - min(d$age.centered) )
plot(table(Ax),xlab="age (centered & shifted)",lwd=4,col=2,
    ylab="frequency")

# sample from our idealized age distribution
f <- seq( from=80 , to=20 , len=34 )
lines( 0:33 , f , lwd=4 , col=1 )
```



Now we can do the simulation with this age distribution.

```
n <- 1000
Asim <- sample(0:33,size=n,replace=TRUE,prob=f)
Asim <- standardize( Asim + min(d$age.centered) )

# sim P(C|do(U=0))
```

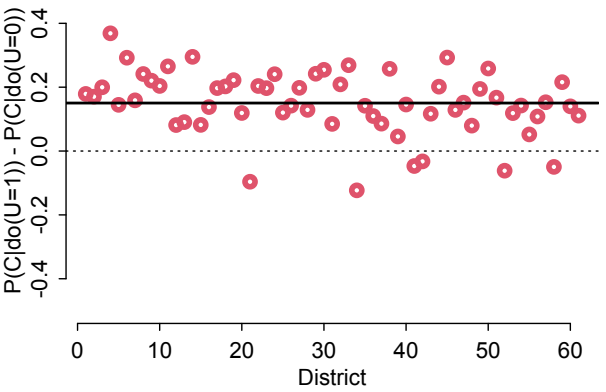
```
pU0 <- sapply( 1:61 ,
    function(dist)
        sim(m4,vars=c("Ks","C"),
            data=list(A=Asim,U=rep(0,n),D=rep(dist,n)))$C
    )

# sim P(C|do(U=1))
pU1 <- sapply( 1:61 ,
    function(dist)
        sim(m4,vars=c("Ks","C"),
            data=list(A=Asim,U=rep(1,n),D=rep(dist,n)))$C
    )
```

And let's plot the contrasts, one for each district:

```
plot( NULL , xlim=c(1,61) , ylim=c(-0.5,0.5) ,
    ylab="P(C|do(U=1)) - P(C|do(U=0))" , xlab="District" )
for ( i in 1:61 ) points( i , mean(pU1[,i] - pU0[,i]) , lwd=4 , col=2 )
abline(h=0,lty=3,lwd=1)

ate <- mean(pU1-pU0)
abline(h=ate,lwd=2)
```



Whew. The treatment impact varies so much, because the districts vary already in rates of C and in the impact of U on C. Age itself doesn't actually do much here. You can adopt another age distribution and see. But getting the calculation right is important, even if a wrong calculation gives the right answer.