STATISTICAL RETHINKING WINTER 2019
HOMEWORK, WEEK 8 SOLUTIONS

1. First, let's set up the data list:

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

dat <- list(
    S = d$surv,
    n = d$density,
    tank = 1:nrow(d),
    pred = ifelse( d$pred=="no" , 0L , 1L ),
    size_ = ifelse( d$size=="small" , 1L , 2L )
)
```

Now to define a series of models. The first is just the varying intercepts model from the text:

```
m1.1 <- ulam(
    alist(
        S ~ binomial( n , p ),
        logit(p) <- a[tank],
        a[tank] ~ normal( a_bar , sigma ),
        a_bar ~ normal( 0 , 1.5 ),
        sigma ~ exponential( 1 )
    ), data=dat , chains=4 , cores=4 , log_lik=TRUE )
```

The other models just incorporate the predictors, as ordinary regression terms.

```
# pred
m1.2 <- ulam(
    alist(
        S ~ binomial( n , p ),
        logit(p) <- a[tank] + bp*pred,
        a[tank] ~ normal( a_bar , sigma ),
        bp ~ normal( -0.5 , 1 ),
        a_bar ~ normal( 0 , 1.5 ),
        sigma ~ exponential( 1 )
    ), data=dat , chains=4 , cores=4 , log_lik=TRUE )

# size
```

```
m1.3 <- ulam(
    alist(
        S ~ binomial( n , p ),
        logit(p) <- a[tank] + s[size_],
        a[tank] ~ normal( a_bar , sigma ),
        s[size_] ~ normal( 0 , 0.5 ),
        a_bar ~ normal( 0 , 1.5 ),
        sigma ~ exponential( 1 )
    ), data=dat , chains=4 , cores=4 , log_lik=TRUE )

# pred + size
m1.4 <- ulam(
    alist(
        S ~ binomial( n , p ),
        logit(p) <- a[tank] + bp*pred + s[size_],
        a[tank] ~ normal( a_bar , sigma ),
        bp ~ normal( -0.5 , 1 ),
        s[size_] ~ normal( 0 , 0.5 ),
        a_bar ~ normal( 0 , 1.5 ),
        sigma ~ exponential( 1 )
    ), data=dat , chains=4 , cores=4 , log_lik=TRUE )

# pred + size + interaction
m1.5 <- ulam(
    alist(
        S ~ binomial( n , p ),
        logit(p) <- a_bar + z[tank]*sigma + bp[size_]*pred + s[size_],
        z[tank] ~ normal( 0 , 1 ),
        bp[size_] ~ normal( -0.5 , 1 ),
        s[size_] ~ normal( 0 , 0.5 ),
        a_bar ~ normal( 0 , 1.5 ),
        sigma ~ exponential( 1 )
    ), data=dat , chains=4 , cores=4 , log_lik=TRUE )
```

I coded the interaction model using a non-centered parameterization. The interaction itself is done by creating a bp parameter for each size value. In this way, the effect of pred depends upon size.

First let's consider the WAIC scores:

```
compare( m1.1 , m1.2 , m1.3 , m1.4 , m1.5 )
```

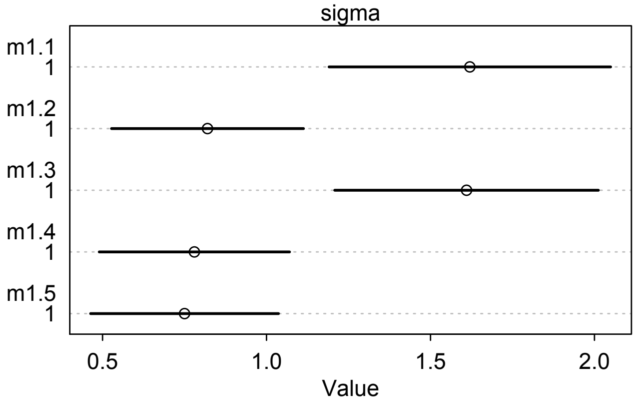
```
      WAIC pWAIC dWAIC weight   SE  dSE
m1.2 198.9  19.1   0.0   0.28 9.06   NA
m1.5 199.3  18.9   0.4   0.23 8.84 3.10
m1.1 199.8  20.7   0.9   0.18 7.22 6.09
m1.3 200.0  20.9   1.2   0.16 7.13 5.99
```

```
m1.4 200.2  19.3   1.4   0.14 8.78 2.03
```

These models are really very similar in expected out-of-sample accuracy. The tank variation is huge. But take a look at the posterior distributions for predation and size. You'll see that predation does seem to matter, as you'd expect. Size matters a lot less. So while predation doesn't explain much of the total variation, there is plenty of evidence that it is a real effect. Remember: We don't select a model using WAIC (or LOO). A predictor can make little difference in total accuracy but still be a real causal effect.

Let's look at all the `sigma` posterior distributions:

```
plot( coeftab( m1.1 , m1.2 , m1.3 , m1.4 , m1.5 ), pars="sigma" )
```



The two models that omit predation, `m1.1` and `m1.3`, have larger values of sigma. This is because predation explains some of the variation among tanks. So when you add it to the model, the variation in the tank intercepts gets smaller.

2. Loading the data and prepping the data list:

```
library(rethinking)
data(bangladesh)
d <- bangladesh
d$district_id <- as.integer(as.factor(d$district))

dat_list <- list(
    C = d$use.contraception,
    did = d$district_id
)
```

Now for the ordinary fixed effect model:

```
m2.1 <- ulam(
    alist(
        C ~ bernoulli( p ),
        logit(p) <- a[did],
        a[did] ~ normal( 0 , 1.5 )
    ) , data=dat_list , chains=4 , cores=4 , log_lik=TRUE )
```

And the varying intercepts model:

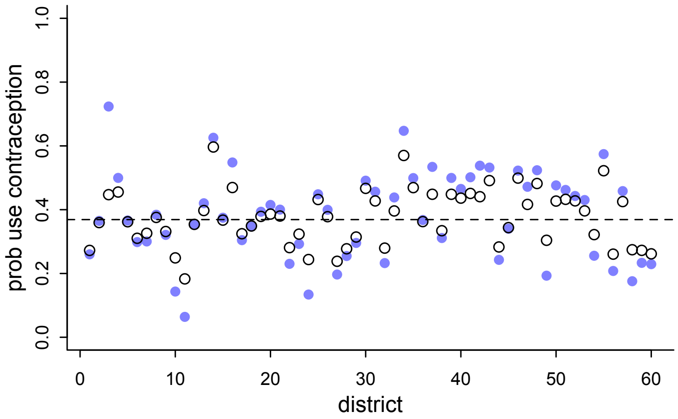
```
m2.2 <- ulam(
    alist(
        C ~ bernoulli( p ),
        logit(p) <- a[did],
        a[did] ~ normal( a_bar , sigma ),
        a_bar ~ normal( 0 , 1.5 ),
        sigma ~ exponential( 1 )
    ) , data=dat_list , chains=4 , cores=4 , log_lik=TRUE )
```

Now let's extract the samples, compute posterior mean probabilities in each district, and plot it all:

```
post1 <- extract.samples( m2.1 )
post2 <- extract.samples( m2.2 )

p1 <- apply( inv_logit(post1$a) , 2 , mean )
p2 <- apply( inv_logit(post2$a) , 2 , mean )

nd <- max(dat_list$did)
plot( NULL , xlim=c(1,nd) , ylim=c(0,1) , ylab="prob use contraception" ,
    xlab="district" )
points( 1:nd , p1 , pch=16 , col=rangi2 )
points( 1:nd , p2 )
abline( h=mean(inv_logit(post2$a_bar)) , lty=2 )
```



The blue points are the fixed estimations. The open points are the varying effects. As you'd expect, they are shrunk towards the mean (the dashed line). Some are shrunk more than others. The third district from the left shrunk a lot. Let's look at the sample size in each district:

```
table(d$district_id)
```

```
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20 
117  20   2  30  39  65  18  37  23  13  21  29  24 118  22  20  24  47  26  15 

 21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40 
 18  20  15  14  67  13  44  49  32  61  33  24  14  35  48  17  13  14  26  41 

 41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60 
 26  11  45  27  39  86  15  42   4  19  37  61  19   6  45  27  33  10  32  42 
```

District 3 has only 2 women sampled. So it shrinks a lot. There are couple of other districts, like 49 and 54, that also have very few women sampled. But their fixed estimates aren't as extreme, so they don't shrink as much as district 3 does.

All of this is explained by partial pooling, of course.

3. First, let's load the data and re-run the old model from Chapter 12:

```
data(Trolley)
d <- Trolley

dat <- list(
    R = d$response,
    A = d$action,
    I = d$intention,
    C = d$contact )

m3.1 <- ulam(
    alist(
        R ~ dordlogit( phi , cutpoints ),
        phi <- bA*A + bC*C + BI*I ,
        BI <- bI + bIA*A + bIC*C ,
        c(bA,bI,bC,bIA,bIC) ~ dnorm( 0 , 0.5 ),
        cutpoints ~ dnorm( 0 , 1.5 )
    ) , data=dat , chains=4 , cores=4 , log_lik=TRUE )
```

Now to run the varying intercept model, we need to build a valid individual ID variable. The IDs in the data are long tags, so we can coerce them to integers in many ways. What is important is that the index values go from 1 to the number of individuals.

```
dat$id <- coerce_index( d$id )
```

Now we can run the model. The only additions here are the `a[id]` in the linear model and the adaptive prior for it.

```
m3.2 <- ulam(
    alist(
        R ~ dordlogit( phi , cutpoints ),
        phi <- a[id] + bA*A + bC*C + BI*I ,
        BI <- bI + bIA*A + bIC*C ,
        a[id] ~ normal( 0 , sigma ),
        c(bA,bI,bC,bIA,bIC) ~ dnorm( 0 , 0.5 ),
        cutpoints ~ dnorm( 0 , 1.5 ),
        sigma ~ exponential(1)
    ) , data=dat , chains=4 , cores=4 , log_lik=TRUE )
```

We can begin by comparing the posterior distributions. The original coefficients are:

```
precis(m3.1)
```

```
     mean   sd  5.5% 94.5% n_eff Rhat
bIC -1.24 0.09 -1.39 -1.09   897 1.01
bIA -0.43 0.08 -0.55 -0.31   828 1.01
bC  -0.34 0.07 -0.45 -0.24  1025 1.00
bI  -0.29 0.06 -0.38 -0.20   774 1.01
bA  -0.47 0.05 -0.55 -0.39   908 1.01
```

And the new ones, having added the individual IDs, are:

```
precis(m3.2)
```

```
       mean   sd  5.5% 94.5% n_eff Rhat
bIC   -1.67 0.10 -1.83 -1.51  1119    1
bIA   -0.56 0.08 -0.69 -0.43  1092    1
bC    -0.45 0.07 -0.57 -0.34  1191    1
bI    -0.39 0.06 -0.48 -0.29  1131    1
bA    -0.65 0.05 -0.73 -0.56  1221    1
sigma  1.91 0.08  1.79  2.04  1831    1
```

Everything has gotten more negative. This is because there is a lot of individual variation in average rating—look at the distribution for `sigma`. That is on the logit scale, so that's a lot of variation on the probability scale. That variation in average rating was hiding some of the effect of the treatments. We get more precision by conditioning on individual.

The WAIC comparison can also help show how much variation comes from individual differences in average rating:

```
compare( m3.1 , m3.2 )
```

```
        WAIC pWAIC  dWAIC weight     SE    dSE
m3.2 31058.2 356.4    0.0      1 179.33     NA
m3.1 36928.9  10.8 5870.7      0  80.72 173.47
```

The WAIC difference is massive. This is consistent with individual variation in average rating being a major effect in this sample.

This is all quite typical of likert-scale data, in my experience. Individuals anchor on different points and this adds noise. When we have repeat samples from the same individual, we can condition away some of that noise and get more precise estimates of the treatment effects.