Lecture 21: Ordered logistic models

** This lecture is based on chapter 11, section 1 of Statistical Rethinking by Richard McElreath.*

```
library(rstan)
library(shinystan)
library(car)
library(mvtnorm)
library(rethinking)
library(MASS)
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
source("../utilityFunctions.R")
```

In the social sciences (and sometimes in biology), it is common to have ordinal response variables—i.e., discrete count-like variables where the value indicates some ordered level along a dimension.

- For example, on a scale of 1–10, how much do you regret you will never get the hours spent in this class back?

When we model this type of data, we need to account for the inherent *ranking/ordering* of the data, because 9 is greater than 8 is greater than 7.

- A big caveat, which is often ignored, is that the distances between levels are not necessarily equal.
  - For example, on teaching evaluations, the distance between fair (3) & poor (2) might be much greater than fair (3) & good (4).
    With such data, what we ultimately want is for any associated predictor variable—as it increases—to move predictions progressively through the categories in sequence.
- For example, if our preference for 80's nostalgia music is positively associated with the number of beers we have had, the model should sequentially move predictions upward as pint number increases.
  - The challenge is mapping the linear model onto the outcomes in the proper order.

The cumulative link

To do this mapping, we will use a *cumulative link function*. The cumulative probability of a value is the probability of any value *or any smaller value*.

Thus the cumulative probability of 3 is the sum of the probabilities of 3, 2, and 1.

- Ordered categories generally start at one, so any result less than one has zero probability.

Motivating example: moral intuition

The data we will use comes from a survey of moral permissiveness conducted by philosophers (Cushman et al. 2006). I am not in love with this example, but it will work.

The experiment relates to "trolley problems." The classic form invokes a runaway trolley or boxcar:

- *Standing by the railroad tracks, Jake sees an empty, out-of-control boxcar about to hit five people. Next to Jake is a lever that can be pulled, sending the boxcar down a side track and away from the five people. But pulling the lever will also lower the railing on a footbridge spanning the side track, causing one person to fall off the footbridge and onto the side track, where he will be hit by the boxcar. If Jake pulls the lever the boxcar will switch tracks and not hit the five people, and the one person to fall and be hit by the boxcar. If Jake does not pull the lever the boxcar will continue down the tracks and hit five people, and the one person will remain safe above the side track.* How morally permissible is it for Jake to pull the lever?

The philosophical quandary is that the qualitative outcome of two scenarios can be identical, but people reach quite different judgements about the morality of the same action in different scenarios.

There are three different principles of unconscious reasoning that might explain variation in judgement:

1. **The action principle:** harm caused by action is morally worse than harm caused by inaction or omission.
2. **The intention principle**: harm intended as the means to an end is morally worse than equivalent harm as an unintentional side effect of a goal.
3. **The contact principle:** using physical contact to cause harm is worse than equivalent harm without using physical contact.

The survey data we will play with varies the above principles, while keeping the same basic story the same. For example, the boxcar story above implies the action principle, but not the others.

A similar story that includes both action and intention is as follows:

- *Standing by the railroad tracks, Heather sees an empty, out-of-control boxcar about to hit five people. Next to Heather is a lever that can be pulled, lowering the railing on a footbridge that spans the main track, and causing one person to fall off the footbridge and onto the main track, where he will be hit by the boxcar. The boxcar will slow down because of the one person, therefore preventing the five from being hit. If Heather pulls the lever the one person will fall and be hit by the boxcar, and therefore the boxcar will slow down and not hit the five people. If Heather does not pull the lever the boxcar will continue down the tracks and hit the five people, and the one person will remain safe above the main track.*

More people judge that Heather pulling the lever is less morally permissible than when Jake pulls the lever as we will see.

```
dat <- read.csv("trolley.csv")
names(dat)
```

```
 [1] "case"      "response"  "order"     "id"        "age"
 [6] "male"      "edu"       "action"    "intention" "contact"
[11] "story"     "action2"
```

```
dim(dat)
```

```
[1] 9930   12
```

```
obs <- dat$response     # response variable
action <- dat$action    # action or no
intent <- dat$intention # intention or no
contact <- dat$contact  # contact or no
```

The dataframe consists of 12 columns and 9930 rows, although multiple rows correspond to one participant. For our purposes, we will ignore pseudoreplication.

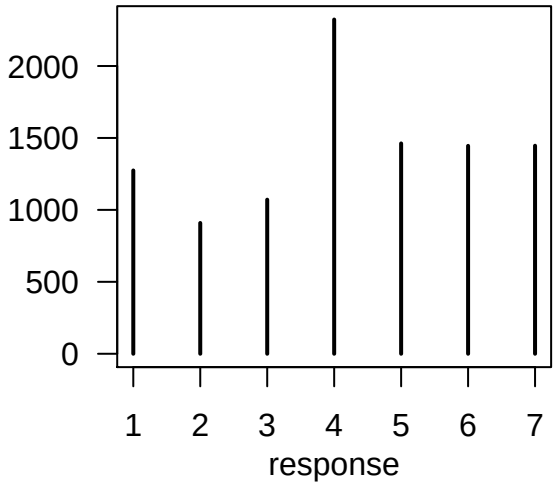
The variables of interest are:

1. `response`: participant's rating of appropriateness of action in story from 1–7 where 1="forbidden", 4="permissible", and 7="obligatory".
2. `action`: treatment code for story with (1) or without (0) action.
3. `intention`: treatment code for intent (1) or not (0).
4. `contact`: treatment code for contact action (1) or not (0).

Describing an ordered distribution with intercepts

To begin, let's look at an overall histogram of the data:

```
par(mar=c(3,3.2,0.1,0.5))
plot(table(obs), type="h", xlim=c(1,7), ann=FALSE, las=1)
mtext("response", side=1, line=2)
```

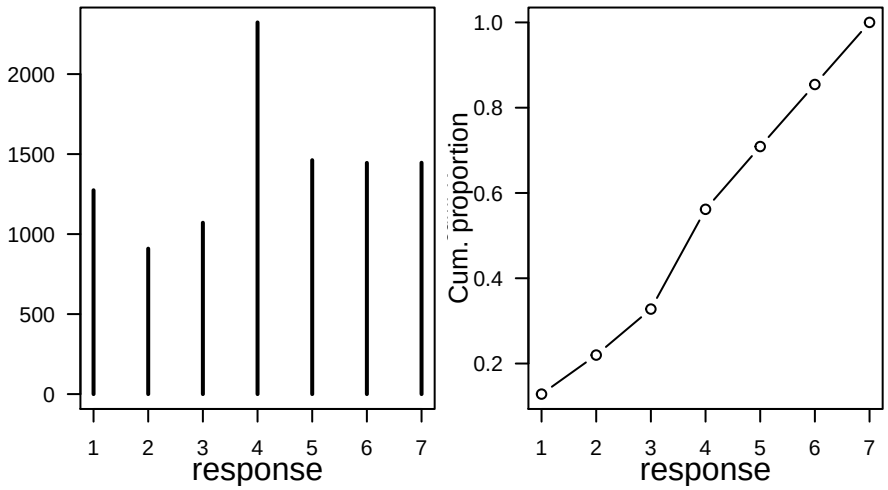


What we want to do is convert this histogram to a log-cumulative-odds scale by constructing the odds of the cumulative probability and then taking the logarithm.

- As with the logit, the cumulative logit is a link function that constrains the probabilities to be in the 0–1 interval.
- When we later add predictors, we can do so safely on the proper probability scale.

```
# discrete proportion of each response variable k
pK <- table(obs)/nrow(dat)

# convert to cumulative proportions
cumPK <- cumsum(pK)
plot(1:7, cumPK, type="b", las=1)
mtext("response", side=1, line=2)
mtext("Cum. proportion", side=2, line=2.2, cex=0.8)
```



The first step, converting to cumulative probabilities (`cumPK`), is shown in the right plot.

To re-describe our histogram as the log-cumulative odds, we will now need intercepts.

- Each intercept will be on the log-cumulative-odds scale and serve as the cumulative probability of each outcome—i.e., applying our link function.

The log-cumulative-odds that our response variable y_i is less than or equal to possible outcome k is:

$$\log \frac{\Pr(y_i \leq k)}{1 - \Pr(y_i \leq k)} = \alpha_k$$

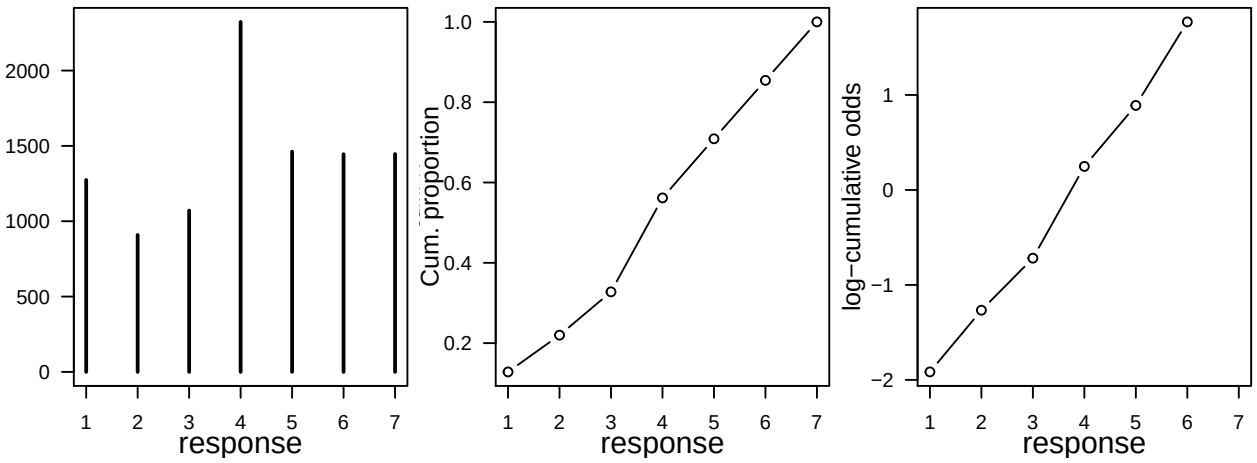
where α_k is the "intercept" unique to each outcome k. These can be directly calculated:

```
logit <- function(x) {
  log(x/(1-x))
}

(lco <- round(logit(cumPK), 4))
```

```
      1       2       3       4       5       6       7
-1.9161 -1.2666 -0.7186  0.2478  0.8899  1.7694     Inf
```

```
plot(1:7, lco, type="b", las=1)
mtext("response", side=1, line=2)
mtext("log-cumulative odds", side=2, line=2.2, cex=0.8)
```



I have plotted these values in the right-most panel. Notice that the largest response $k_7 = \infty$. This is because $\log(1/(1-1)) = \infty$.

- The largest category always has a cumulative probability of 1, so we don't need a parameter for it.
- For $K = 7$ possible response values, we only need $K - 1 = 6$ intercepts.

Bayesian implementation

The above wankery is all well and good, but it would be nice to model the uncertainty in our estimates as well as (later) add in predictors.

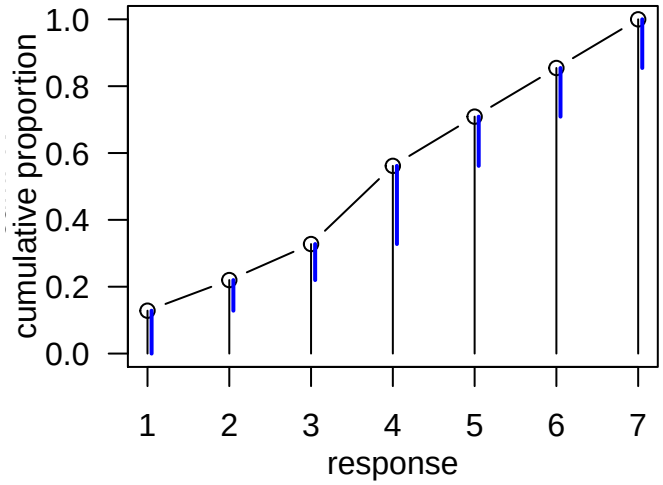
This means we need a likelihood function of each response variable to go from the cumulative probabilities, $\Pr(y_i \leq k)$, to $\Pr(y_i = k)$.

```
par(mar=c(3,3.2,0.1,0.5))

plot(cumPK, type="h", xlim=c(1,7), ylim=c(0,1), ann=FALSE, las=1)

par(new=TRUE)
plot(cumPK, type="b", axes=FALSE, ylim=c(0,1))
segments(x0=1+0.05, x1=1+0.05, y0=0, y1=cumPK[1], col="blue", lwd=2)

for(i in 2:7)
  segments(x0=i+0.05, x1=i+0.05, y0=cumPK[i-1], y1=cumPK[i],
    col="blue", lwd=2)
mtext("response", side=1, line=2)
mtext("cumulative proportion", side=2, line=2.2)
```



This figure shows how we accomplish our goal. Each intercept α_k describes a cumulative probability for each k. So we just need the inverse link function to translate from log-cumulative odds back to cumulative probability.

When we observe k and need it's likelihood, we get that likelihood by subtraction:

$$p_k = \Pr(y_i = k) = \Pr(y_i \leq k) - \Pr(y_i \leq k - 1).$$

The blue line segments in our figure are the likelihoods for each k, calculated by subtraction.

Let's specify our model now:

$$
\begin{aligned}
y_i &\sim \text{Ordered Logistic}(\mathbf{p}) \\
\text{logit}(p_k) &= \alpha_k \\
\alpha_k &\sim \text{Normal}(0, \sigma_\alpha).
\end{aligned}
$$

The ordered logistic distribution is simply a *categorical* distribution—a special case of a multinomial where $N = 1$—that takes a vector of probabilities of each response variable below the maximum response.

Here is the Stan model:

```
data{
  int<lower=0> nObs;      // No. obs.
  int<lower=0> K;         // max no. categories
  int<lower=1> obs[nObs]; // obs.
  real<lower=0> alphaSD;  // prior for alpha SD
}

parameters{
  ordered[K-1] alpha;   // intercepts
}

model{
  vector[nObs] phi;

  alpha ~ normal(0 , alphaSD);
  for ( i in 1:nObs ) {
    phi[i] = 0;
    obs[i] ~ ordered_logistic( phi[i] , alpha );
  }
}
```

Here, ϕ is a placeholder for a linear model that incorporates predictors. Right now it is just a constant zero because only the intercepts are of interest.

```
nObs <- nrow(dat)
K <- max(obs)

# function to set initial values so the ordering is correct
inits <- function() {
  list(alpha = sort(runif(K-1, -2,2)))
}

d <- list(nObs=nObs, K=K, obs=obs, alphaSD=5)
m1 <- stan(file="21.simpOrd.stan", data=d, iter=2000, chains=4,
  seed=867.5309, init=inits)
```

```
print(m1, pars="alpha")
```

```
Inference for Stan model: 21.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

          mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
alpha[1] -1.92       0 0.03 -1.98 -1.94 -1.92 -1.90 -1.86  2977    1
alpha[2] -1.27       0 0.02 -1.31 -1.28 -1.27 -1.25 -1.22  4000    1
alpha[3] -0.72       0 0.02 -0.76 -0.73 -0.72 -0.70 -0.68  4000    1
alpha[4]  0.25       0 0.02  0.21  0.23  0.25  0.26  0.29  4000    1
alpha[5]  0.89       0 0.02  0.85  0.88  0.89  0.91  0.93  4000    1
alpha[6]  1.77       0 0.03  1.71  1.75  1.77  1.79  1.83  4000    1

Samples were drawn using NUTS(diag_e) at Tue May  1 16:13:19 2018.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

The first thing to see is that, because we had a lot of data, our estimates of α are precise (look at the tiny SDs).

To get our cumulative probabilities back, we just take the inverse logit of our results:

```
alpha <- as.data.frame(m1, pars="alpha")
(probs <- logistic(colMeans(alpha)))
```

```
 alpha[1]  alpha[2]  alpha[3]  alpha[4]  alpha[5]  alpha[6]
0.1281605 0.2197434 0.3276633 0.5616996 0.7090363 0.8545445
```

Adding predictors

Thus far we have essentially figured out how to make a Bayesian histogram. Now we we should include some predictor variables.

To include them, we will define the log-cumulative odds of each response k as a sum of it's intercept and a linear model.

If we want to add x to the model, our linear portion of the model becomes $\phi_i = \beta x_i$. Then the cumulative logit becomes:

$$\log \frac{\Pr(y_i \leq k)}{1 - \Pr(y_i \leq k)} = \alpha_k - \phi_i$$
$$\phi_i = \beta x_i$$

This keeps the correct ordering of response values, while still changing the likelihood of each individual value as the predictors x_i change.

We subtract the linear model from the intercept because decreasing the log-cumulative odds of each outcome k below the maximum shifts probability mass upward towards higher values.

As an example, we can use the `rethinking` function `dordlogit` to take our mean estimates of α from `m1` and subtract 0.5 from each:

```
# original estimates:
(pk <- round(dordlogit(1:K, phi = 0, colMeans(alpha)), 3))
```

```
[1] 0.128 0.092 0.108 0.234 0.147 0.146 0.145
```

```
# average outcome
sum(pk*1:7)
```

```
[1] 4.198
```

```
# subtracting 0.5 from each:
(pk <- round(dordlogit(1:K, phi = 0.5, colMeans(alpha)), 3))
```

```
[1] 0.082 0.064 0.082 0.209 0.159 0.184 0.219
```

```
sum(pk*1:7)
```

```
[1] 4.724
```

Thus, a positive β value indicates that an increase in a predictor variable results in an increase in the average response.

Now, we can create a model that includes predictor variables for the different principles (`action`, `intent`, & `contact`).

Also, because stories included multiple principles at once, we might expect interactions between `action` & `intent` and `intent` & `contact`.

Therefore our model is now:

$$\log \frac{\Pr(y_i \leq k)}{1 - \Pr(y_i \leq k)} = \alpha_k - \phi_i$$
$$\phi_i = \beta_A A_i + \beta_I A_i + beta_C C_i + \beta_{AI} A_i I_i + \beta_{IC} I_i C_i$$

```
X <- model.matrix(~action*intent + intent*contact)[,-1]
nVar <- ncol(X)

inits2 <- function() {
  list(alpha = sort(runif(K-1, -2,2)),
       beta = runif(nVar, -2,2))
}
```

```
d2 <- list(nObs=nObs, K=K, nVar=nVar, obs=obs, X=X, alphaSD=5,
  betaSD=1)
m2 <- stan(file="21.linearOrd.stan", data=d2, iter=1000, chains=4,
  seed=867.5309, init=inits2)
```

```
print(m2, pars=c("alpha", "beta"))
```

```
Inference for Stan model: 21.
4 chains, each with iter=1000; warmup=500; thin=1;
post-warmup draws per chain=500, total post-warmup draws=2000.

          mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
alpha[1] -2.64       0 0.05 -2.74 -2.67 -2.64 -2.60 -2.54  1022 1.00
alpha[2] -1.94       0 0.05 -2.03 -1.97 -1.94 -1.91 -1.85  1053 1.01
alpha[3] -1.34       0 0.04 -1.43 -1.37 -1.34 -1.32 -1.26  1015 1.01
alpha[4] -0.31       0 0.04 -0.40 -0.34 -0.31 -0.28 -0.22  1033 1.01
alpha[5]  0.36       0 0.04  0.28  0.33  0.36  0.39  0.45  1047 1.01
alpha[6]  1.27       0 0.05  1.17  1.24  1.27  1.30  1.36  1172 1.00
beta[1]  -0.47       0 0.05 -0.58 -0.51 -0.47 -0.44 -0.37  1056 1.01
beta[2]  -0.29       0 0.06 -0.40 -0.32 -0.29 -0.25 -0.18   924 1.00
beta[3]  -0.33       0 0.07 -0.47 -0.38 -0.34 -0.29 -0.19  1125 1.01
beta[4]  -0.44       0 0.08 -0.60 -0.49 -0.44 -0.39 -0.30  1163 1.00
beta[5]  -1.26       0 0.10 -1.45 -1.33 -1.27 -1.20 -1.08  1095 1.00

Samples were drawn using NUTS(diag_e) at Tue May  1 17:38:26 2018.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

So how do we interpret our estimates? The $\alpha's$ are interpreted as before, but are not much use without taking the inverse logit to get the cumulative probabilities.

The slopes β are all negative. This means that each factor/interaction decreases the average response. Thus including any of the principles means people find them less morally permissible.

Unfortunately, we can't go much further with tables alone, because the slopes, even if changed to cumulative probabilities, apply to every value of the response (except the 7th because the cumulative odds are fixed at ∞).

So let's make some interaction plots:

```
#extract the results:
post <- as.data.frame(m2, pars=c("alpha", "beta"))
names(post) <- c("a1", "a2", "a3", "a4", "a5", "a6",
  "bA", "bI", "bC", "bAI", "bIC")
```

```r
par(mar=c(3,3.2,1.2,0.5))
par(mfrow=c(1,3))
xseq <-seq(0.9, 0.1, length=7)

# empty plot for when A=C=0
plot(0:1,0:1, type="n", xlim=c(0,1), ylim=c(0,1), xaxp=c(0,1,1), las=1)
mtext("interaction", side=1, line=2)

A <- 0  # value for action
C <- 0  # value for contact
I <- 0:1 # value for interaction

for(i in 1900:2000) {
  pz <- post[i,]
  ak <- as.numeric(pz[1:6])
  phi <- pz$bA*A + pz$bI*I + pz$bC*C + pz$bAI*A*I + pz$bIC*I*C
  pk <- pordlogit(1:6, a=ak, phi=phi)

  for(k in 1:6) lines(I, pk[,k], col="#6495ed30")
}
abline(h=0:1, lty=2)
mtext(concat("action=", A, ", contact=",C))
text(x=xseq, y=c(0.03, 0.12, 0.2, 0.35, 0.53, 0.7, 0.9),
  labels = 1:7, col="blue", cex=1.1)


# empty plot for A=1, C=0
plot(0:1,0:1, type="n", xlim=c(0,1), ylim=c(0,1), xaxp=c(0,1,1), las=1)
mtext("interaction", side=1, line=2)

A <- 1  # value for action
C <- 0  # value for contact
I <- 0:1

for(i in 1900:2000) {
  pz <- post[i,]
  ak <- as.numeric(pz[1:6])
  phi <- pz$bA*A + pz$bI*I + pz$bC*C + pz$bAI*A*I + pz$bIC*I*C
  pk <- pordlogit(1:6, a=ak, phi=phi)

  for(k in 1:6) lines(I, pk[,k], col="#6495ed30")
}
abline(h=0:1, lty=2)
mtext(concat("action=", A, ", contact=",C))
```

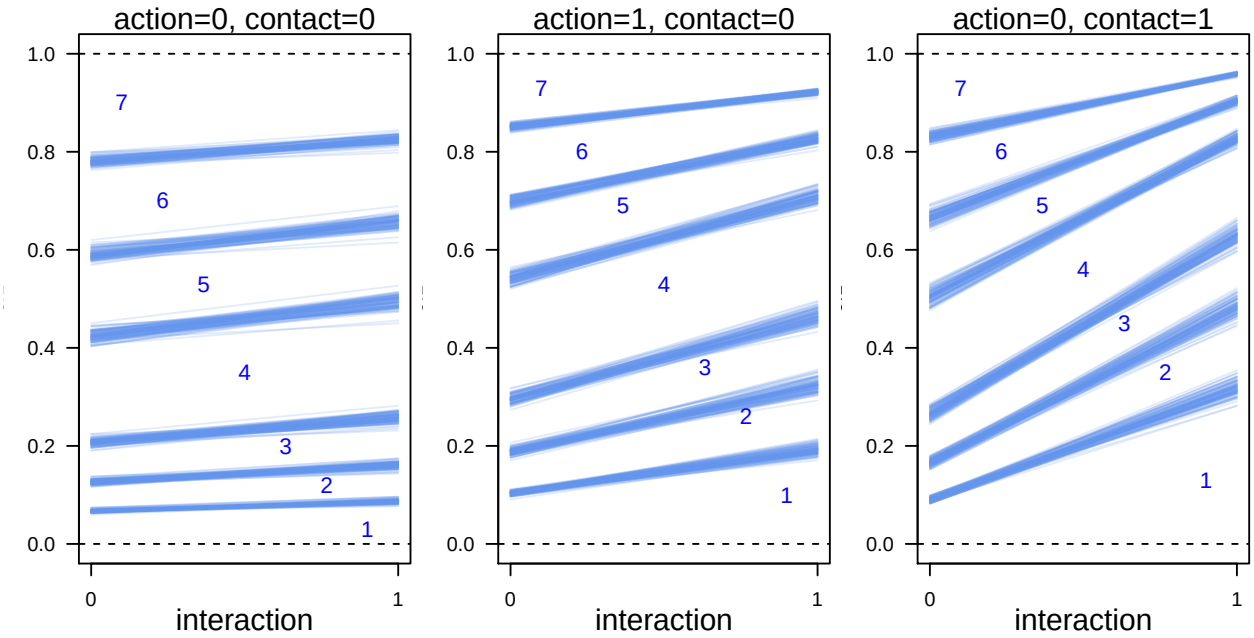
```
text(x=xseq, y=c(0.1, 0.26, 0.36, 0.53, 0.69, 0.8, 0.93),
  labels = 1:7, col="blue", cex=1.1)


# empty plot for A=0, C=1
plot(0:1,0:1, type="n", xlim=c(0,1), ylim=c(0,1), xaxp=c(0,1,1), las=1)
mtext("interaction", side=1, line=2)

A <- 0  # value for action
C <- 1  # value for contact
I <- 0:1

for(i in 1900:2000) {
  pz <- post[i,]
  ak <- as.numeric(pz[1:6])
  phi <- pz$bA*A + pz$bI*I + pz$bC*C + pz$bAI*A*I + pz$bIC*I*C
  pk <- pordlogit(1:6, a=ak, phi=phi)

  for(k in 1:6) lines(I, pk[,k], col="#6495ed30")
}
abline(h=0:1, lty=2)
mtext(concat("action=", A, ", contact=",C))
text(x=xseq, y=c(0.13, 0.35, 0.45, 0.56, 0.69, 0.8, 0.93),
  labels = 1:7, col="blue", cex=1.1)
```



What I have done is looped through the last 100 iterations of the posterior and plotted the cutoff points. Each plot is holding `action` and `contact` constant while plotting the change

in interaction.

We can interpret our results by looking at the white space in between the blue lines. The larger the white space, the greater that value of k.

For example, in the last plot, including `interact` in the story problem shifts the probability mass downward such that the "forbidden" (1) category has the highest probability.

References

Cushman et al. 2006. Psychological Science 17:1082–1089.

McElreath (2016). Statistical Rethinking: A Bayesian Course with Examples in R and Stan. CRC Press.