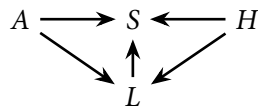


STATISTICAL RETHINKING 2025

WEEK 9 SOLUTIONS

1. Before we get started on modeling a function to relate age to hunting success, let's think about this problem causally. Age could work through multiple causal paths. We want to estimate all of them. From what I've given you, the causes to consider are age (A), trip duration (L), and unmeasured features of each hunter (H). A minimalist view of how these causes interact to influence success (S) might be:



So the total causal effect of age acts partly through any influence it has on the duration of each trip, if in fact duration has much influence on success. So we do not want to stratify by duration (L). This means we can ignore the missing values in L , because we don't need to include L in the model. Isn't that nice? (We probably should stratify by hunter, but we'll do that in a later problem.)

The problem now is find some reasonable function to describe the causal effect of age on success. Remember: Getting the function right is as important as getting the arrows right. I am going to work my way up to a satisfactory age function. A linear model is obviously not a good starting place, because any age-related increases in success must taper off at some point—80 year olds can't be getting more skilled at the same rate they did when they were 20. So I'll try $\log(\text{age})$, which imposes diminishing returns in an automatic way. Here's the model, just a bernoulli model:

```
library(rethinking)
data(Achehunting)
d <- Achehunting
dat <- list(
  S = ifelse(d$kg.meat>0,1,0),
  A = standardize(log(d$age)) )

# log age model
m1a <- ulam(
  alist(
    S ~ bernoulli(p),
    logit(p) <- a + bA*A,
    a ~ normal(0,1),
    bA ~ normal(0,0.5)
  ) , data=dat , chains=4 , cores=4 )
precis(m1a)
```

```

      mean    sd  5.5% 94.5% rhat ess_bulk
a  -0.09 0.02 -0.12 -0.06 1.00  1318.05
bA  0.12 0.02  0.09  0.14 1.01  1393.43

```

The precis output is not so useful, but you can check the diagnostics and make a quick assessment that the coefficient on $\log(\text{age})$ is positive, indicating success increases with age.

Let's plot the posterior function though. It can be hard to plot posterior predictions for Bernoulli data, because all the observations are at 0 or 1. So instead let's bin the age values and compute proportions of success in each bin. This will do it:

```

# round age to nearest decade
Ar <- round( d$age / 10 ) * 10
Aseq <- c(10,20,30,40,50,60,70,80)
SA <- sapply( Aseq , function(a) mean(dat$S[Ar==a]) )
plot( Aseq , SA , ylim=c(0,1) , xlim=c(0,90) , type="b" , lwd=3 ,
      col=2 , xlab="age (years)" , ylab="proportion successful" )

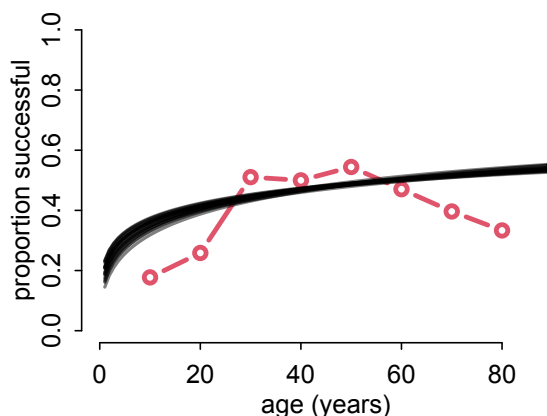
```

And now to overlay the posterior function (20 draws from posterior). This code is only awkward because the plot has years on the horizontal, but the model was fit on standardized log years. So I need to do that adjustment instead the plotting code below.

```

post <- extract.samples(m1a)
for ( i in 1:20 )
  curve( inv_logit(
    post$a[i] + post$bA[i]*( log(x)-3.79 )/0.335 ) ,
    add=TRUE , lwd=2 , col=grau() , from=1 )

```



This is surely one of the worst fits I have ever produced. What is needed is something that isn't monotonic, something that can rise and then fall. A simply polynomial would work. But you know I am not a fan of polynomials. So I'm going to revive

one of those growth models that I used in a previous homework, when I modeled dinosaur growth. Let the probability of success at an age A be:

$$p(A) = \alpha(1 - \exp(-\beta_1 A)) \exp(-\beta_2 A)$$

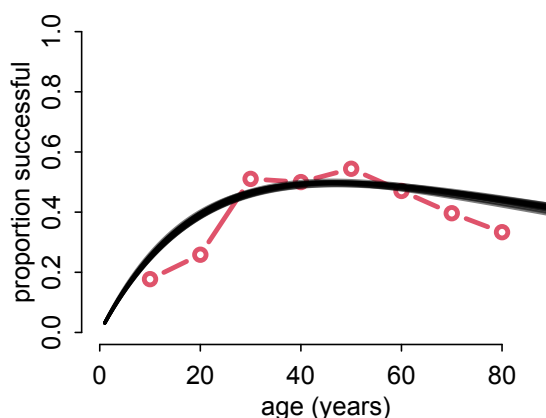
The parameter α sets the maximum, so we constrain it to be between zero and one. The β parameters control the rates of increase (β_1) and decrease (β_2).

Now it's just a simple matter of sticking this function into the model. We also need to add age on the natural scale to the data. I'll normalize it though by dividing by a reference age of 80. This just makes the fitting easier, because the coefficients won't multiply such large values.

```
dat$A2 <- d$age / 80
m1b <- ulam(
  alist(
    S ~ bernoulli(p),
    p <- a*(1-exp(-b1*A2))*exp(-b2*A2),
    a ~ beta(4,4),
    c(b1,b2) ~ exponential(2)
  ) , data=dat , chains=4 , cores=4 )
```

Plotting:

```
post <- extract.samples(m1b)
for ( i in 1:20 ) with( post ,
  curve( a[i]*(1-exp(-b1[i]*x/80))*exp(-b2[i]*x/80) ,
    add=TRUE , lwd=2 , col=grau() , from=1 ) )
```



Much better. Still looks like it rises too quickly at the start. Do babies really learn to hunt so quickly? We could fix this by shifting the age where the functions goes to zero. Or we could impose some increasing returns on the increasing portion of the

function. Let's try that. Add one more parameter:

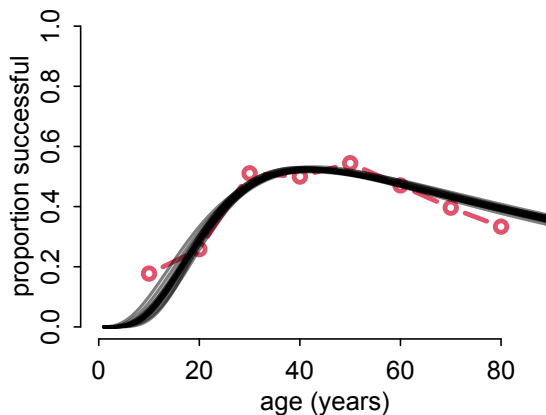
$$p(A) = \alpha(1 - \exp(-\beta_1 A))^\gamma \exp(-\beta_2 A)$$

I've inserted a γ , which manages the “elasticity” of the term that increases with age. If $\gamma > 1$, then the curve should accelerate at first. Let's try fitting it:

```
m1c <- ulam(
  alist(
    S ~ bernoulli(p),
    p <- a*exp(-b2*A2)*(1-exp(-b1*A2))^g,
    a ~ beta(4,4),
    g ~ exponential(0.5),
    c(b1,b2) ~ exponential(2)
  ) , data=dat , chains=4 , cores=4 )
```

Plotting:

```
post <- extract.samples(m1c)
for ( i in 1:20 ) with( post ,
  curve( a[i]*exp(-b2[i]*x/80)*(1-exp(-b1[i]*x/80))^g[i] ,
    add=TRUE , lwd=2 , col=grau() , from=1 ) )
```



That is much more satisfying. This has been a curve-fitting exercise. So it can be interesting to compare the PSIS scores of these models. To do so, run each model again after adding `log_lik=TRUE` to the `ulam` call. I do not do that by default, because there are 14-thousand observations here and `log_lik` produces a full posterior for each observation. That's a lot of output. But go ahead and do it and be patient with your computer. Then:

```
compare( m1a , m1b , m1c , func=PSIS )
```

```
PSIS    SE dPSIS    dSE pPSIS weight
```

```
m1c 19590.7 32.02 0.0 NA 3.0 1
m1b 19704.7 16.96 114.0 18.09 1.8 0
m1a 19838.6 17.20 247.8 27.15 1.9 0
```

This echoes what you can see in the posterior prediction plots, that the third model does much better.

And the story it tells is that success increases rapidly during the teens and then declines slowly after middle age. Note also that the success rate never rises much above 50% at its peak. Hunting is hard.

It's reasonable now to ask whether we've overfit, because we chose the function by iteratively looking at model fits. It's possible. The function is defensible in terms of its biological features. But if we had a different sample, maybe we would have chosen to model other biological features. Research is iterative. Just be transparent about it.

2. I will let the two rate parameters, β_1 and β_2 , vary by individual hunter. So we need two varying effects. I'll use a non-centered parameterization. But I'll also add some code to convert them to centered effects, to keep the linear model simple. The trick to adding varying effects to this model is to maintain the constraints on β_1 and β_2 . They both have to be positive. So when we draw varying effects from a normal distribution, we have to then transform them to be positive before using them in the linear model. This means in practice that we set the varying effects on the log scale and then exp them before use.

Here's the code. This model might take a little while to run. It took about 20 minutes on my laptop. Start it up and then go for a walk.

```
# hunter index variable
dat$H <- as.integer(as.factor(d$id))
dat$NH <- max(dat$H)

m2 <- ulam(
  alist(
    S ~ bernoulli(p),
    p <- a*exp(-b2H[H]*A2)*(1-exp(-b1H[H]*A2))^g,
    # centered varying effects
    transpars> vector[NH]:b1H <- exp(b1+V[1:NH,1]),
    transpars> vector[NH]:b2H <- exp(b2+V[1:NH,2]),
    # non-centered varying effects
    transpars> matrix[NH,2]:V <-
      compose_noncentered( sigma_H , L_Rho_H , Z ),
    matrix[2,NH]:Z ~ normal( 0 , 1 ),
    cholesky_factor_corr[2]:L_Rho_H ~ lkj_corr_cholesky( 4 ),
    vector[2]:sigma_H ~ exponential(1),
    # fixed priors
    a ~ beta(4,4),
    g ~ exponential(0.5),
    c(b1,b2) ~ normal(0,0.5),
```

```
gq> matrix[2,2]:Rho_H <- Chol_to_Corr( L_Rho_H )
) , data=dat , chains=4 , cores=4 , iter=4000 )
precis(m2,3,pars=c("a","g","b1","b2","sigma_H"))
```

	mean	sd	5.5%	94.5%	rhat	ess_bulk
a	0.90	0.04	0.83	0.96	1	4257.00
g	8.07	2.19	4.99	11.88	1	1660.29
b1	2.23	0.12	2.02	2.42	1	1122.56
b2	0.04	0.12	-0.17	0.21	1	2022.50
sigma_H[1]	0.17	0.08	0.04	0.31	1	653.82
sigma_H[2]	0.59	0.09	0.47	0.74	1	827.08

The variance components suggest that there is little variation in β_1 but more in β_2 . So hunters differ more in their decline rates than their increase rates. Of course a faster decline rate will lower the entire curve for a hunter, so it effectively changes the maximum as well.

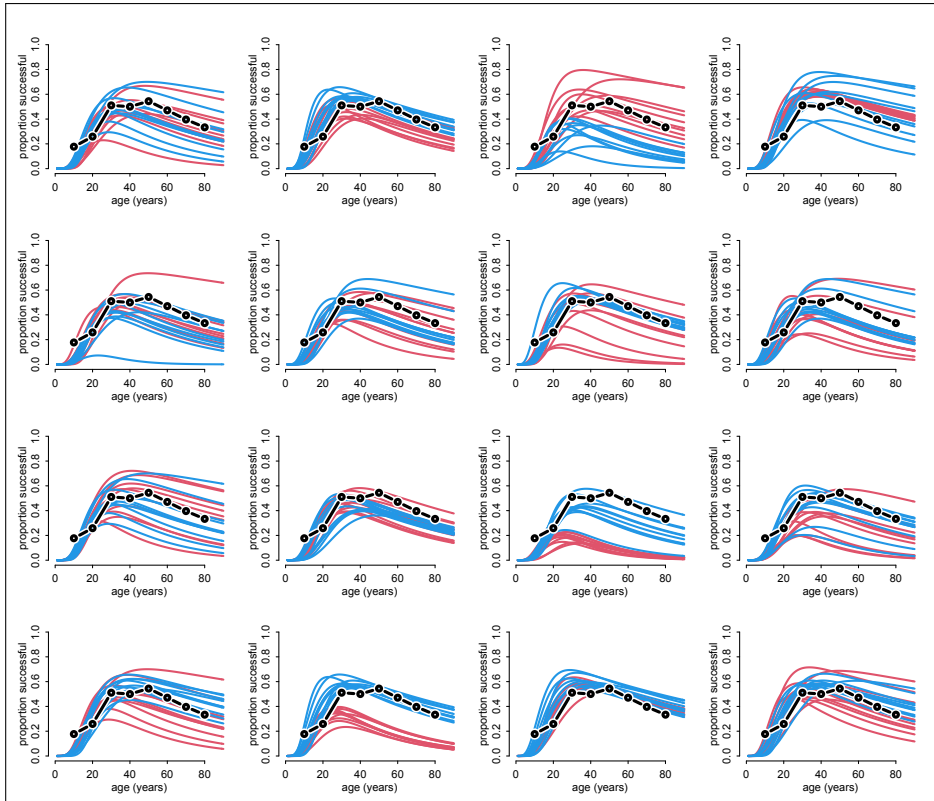
To get an impression for how much variation there is among hunters, let's plot some hunter functions. I'll make a grid of 16 plots, each showing a pair of hunters, with samples from the posterior shown in different colors for each hunter. We can compare within plots and across plots.

```
post <- extract.samples(m2)

# round age to nearest decade
Ar <- round( d$age / 10 ) * 10
Aseq <- c(10,20,30,40,50,60,70,80)
SA <- sapply( Aseq , function(a) mean(dat$S[Ar==a]) )

par(mfrow=c(4,4))
for (k in 1:16 ) {
  plot( Aseq , SA , ylim=c(0,1) , xlim=c(0,90) , type="b" ,
        lwd=3 , col=0 , xlab="age (years)" , ylab="proportion successful" )

  # plot nj random hunters
  cols <- c(2,4,5,6)
  nj <- 2
  hseq <- sample(1:dat$NH,size=nj)
  for ( j in hseq )
    for ( i in 1:10 ) with( post ,
      curve( a[i]*exp(-b2H[i,j]*x/80)*(1-exp(-b1H[i,j]*x/80))^g[i] ,
            add=TRUE , lwd=2 , from=1 , col=cols[which(hseq==j)] ) )
  points( Aseq , SA , type='b' , col="white" , lwd=6 )
  points( Aseq , SA , type='b' , col=1 , lwd=3 )
}#k
```



I show the plot on the next page. You can run the code above over and over again to sample new hunters. The overall impression is that hunters can vary quite a lot. But many of them are well represented by the mean.

The question asked if there is more variation by age or by hunter. The easy answer is that there is lots of both. But a principled answer (that is harder) is to compute the variation in success across age, averaged over hunters, and compare it to the variation across hunters, averaged over age.

This is a subtle kind of calculation to do, because just using every age in the data is a bit weird—many hunters will never live to 80, so weighting the estimates at 80 years old the same as those at 20 years old seems wrong. Indeed it is wrong, if the goal is to say how much empirical realized variation in success owes to age differences or to other individual differences. Suppose for example that there is only one hunter who is 90 in the community. But he is really really good. He still won't contribute much to variation in success as explained by age differences in the total sample, because he is just one person.

Let's do it the easy way first, weighting each age the same. This isn't ideal, but it is the simplest example. First I compute the variation across age for each hunter.

```
# compute variation across age for each hunter
vHA <- rep(NA,dat$NH)
Aseq <- 10:80
for ( i in 1:dat$NH ) {
  # compute variation across age for each sample from posterior
  # then average across samples
  # v has margins [samples,ages]
  v <- sapply( Aseq , function(x)
    with( post ,
      a*exp(-b2H[,i]*x/80)*(1-exp(-b1H[,i]*x/80))^g ) )
  # now average variation across ages
  vHA[i] <- mean( apply( v , 1 , var ) )
}#i
```

Now the variation across hunters at each age.

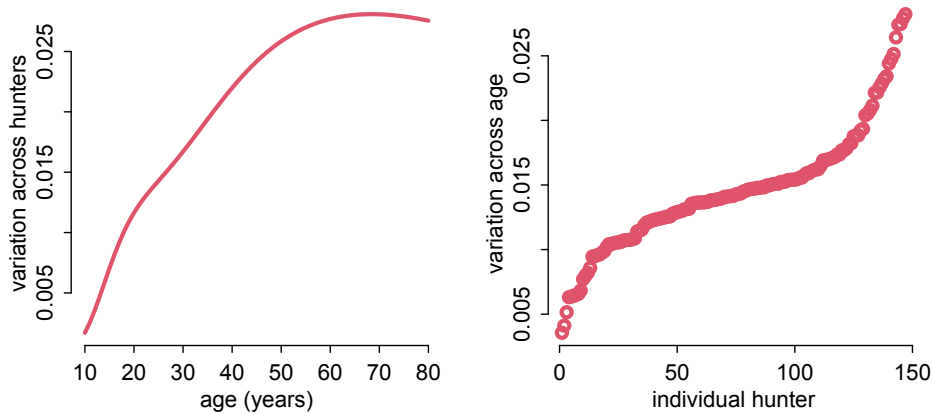
```
# variation across all hunters at each age
vAH <- rep(NA,length(Aseq))
for ( j in 1:length(Aseq) ) {
  # variation at age j across all hunters
  # v has margins [samples,hunters]
  v <- sapply( 1:dat$NH , function(i)
    with( post ,
      a*exp(-b2H[,i]*Aseq[j]/80)*(1-exp(-b1H[,i]*Aseq[j]/80))^g ) )
  # average variance across individuals
  vAH[j] <- mean( apply( v , 1 , var ) )
}#j
```

And we can compare these calculations directly:

```
# average across hunters (of variation across age)
mean(vHA)
# average across age (of variation across hunters)
mean(vAH)
```

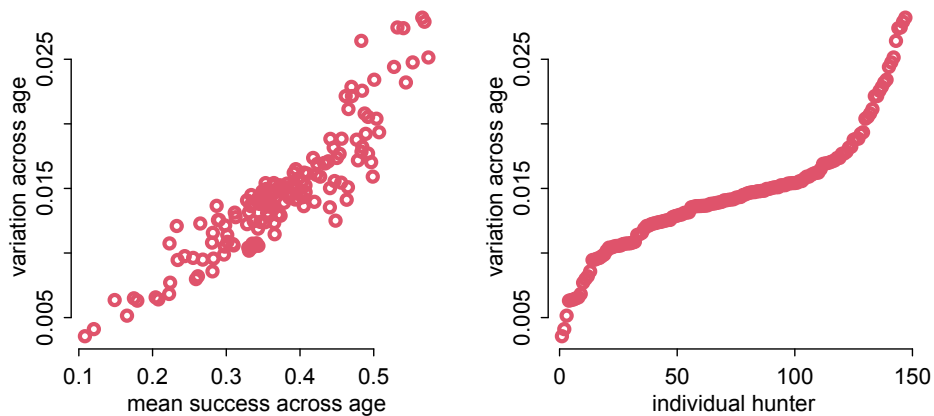
```
> mean(vHA)
[1] 0.01458467
> mean(vAH)
[1] 0.02096086
```

So the variation across ages is about 40% greater than that across individuals. Let's plot the elements of these calculations, to get a sense of what contributes to this difference.



The left shows the average variation at each age across hunters. Differences across hunters are greatest late in life, which is consistent with the variance components showing greater variation for the decline rates. Since late ages contribute the most variation across hunters, if we redid this comparison discounting later ages, since they appear less in the community, then the variation explained by individual differences might be more similar or even greater than the variation across age.

The right is each hunter on the horizontal and his posterior mean variation across ages. Some hunters change a lot over time, especially the good ones. This is easier to see if I plot the variation across age against the mean success across age:



The best hunters also vary the most over a lifetime.