STATISTICAL RETHINKING 2022
WEEK 4 SOLUTIONS

1. I won't repeat the models here. They are in the text. Model `m6.9` contains both marriage status and age. Model `m6.10` contains only age. Model `m6.9` produces a confounded inference about the relationship between age and happiness, due to opening a collider path. To compare these models using PSIS and WAIC:

```
compare( m6.9 , m6.10 , func=PSIS )
compare( m6.9 , m6.10 , func=WAIC )
```

```
        PSIS    SE dPSIS  dSE pPSIS weight
m6.9  2714.0 37.57   0.0   NA   3.8      1
m6.10 3101.9 27.76 387.9 35.4   2.4      0

        WAIC    SE dWAIC   dSE pWAIC weight
m6.9  2714.3 37.51   0.0    NA   3.9      1
m6.10 3101.9 27.68 387.6 35.34   2.3      0
```

The model that produces the invalid inference, `m6.9`, is expected to predict much better. And it would. This is because the collider path does convey actual association. We simply end up mistaken about the causal inference. We should not use PSIS or WAIC to choose among models, unless we have some clear sense of the causal model. These criteria will happily favor confounded models.

So what about the coefficients in the confounded model?

```
precis( m6.9 , depth=2 )
```

```
       mean   sd  5.5% 94.5%
a[1]  -0.24 0.06 -0.34 -0.13
a[2]   1.26 0.08  1.12  1.39
bA    -0.75 0.11 -0.93 -0.57
sigma  0.99 0.02  0.95  1.03
```

We cannot interpret these estimates without reference to the causal model. So let's remember that the causal model is just:

$$H \rightarrow M \leftarrow A$$

where H is happiness, M is married, and A is age.

Okay, you know that the `bA` parameter is bias by the collider relationship. This model suffers from collider bias, and so `bA` is not anything but a conditional association. It isn't any kind of causal effect.

The parameters `a[1]` and `a[2]` are intercepts for unmarried and married, respectively. But do they correctly estimate the effect of marriage on happiness? No, because marriage in this example does not influence happiness. It is a consequence of happiness.

So what do they estimate? They measure the association between marriage and happiness. But they do it with bias, because the model also includes age. To prove this to yourself, fit a model that stratifies happiness by marriage status but ignore age. You'll see that the `a[1]` and `a[2]` estimates you get are different, once you omit age from the model.

In sum, every parameter in the model is a non-causal association.

2. Here are five models with different combinations of predictors:

```
library(rethinking)
data(foxes)
d <- foxes
d$W <- standardize(d$weight)
d$A <- standardize(d$area)
d$F <- standardize(d$avgfood)
d$G <- standardize(d$groupsize)

m1 <- quap(
    alist(
        W ~ dnorm( mu , sigma ),
        mu <- a + bF*F + bG*G + bA*A,
        a ~ dnorm(0,0.2),
        c(bF,bG,bA) ~ dnorm(0,0.5),
        sigma ~ dexp(1)
    ), data=d )
m2 <- quap(
    alist(
        W ~ dnorm( mu , sigma ),
        mu <- a + bF*F + bG*G,
        a ~ dnorm(0,0.2),
        c(bF,bG) ~ dnorm(0,0.5),
        sigma ~ dexp(1)
    ), data=d )
m3 <- quap(
    alist(
        W ~ dnorm( mu , sigma ),
```

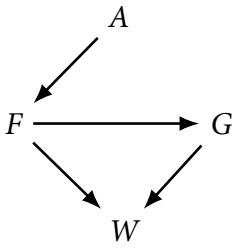
```
        mu <- a + bG*G + bA*A,
        a ~ dnorm(0,0.2),
        c(bG,bA) ~ dnorm(0,0.5),
        sigma ~ dexp(1)
    ), data=d )
m4 <- quap(
    alist(
        W ~ dnorm( mu , sigma ),
        mu <- a + bF*F,
        a ~ dnorm(0,0.2),
        bF ~ dnorm(0,0.5),
        sigma ~ dexp(1)
    ), data=d )
m5 <- quap(
    alist(
        W ~ dnorm( mu , sigma ),
        mu <- a + bA*A,
        a ~ dnorm(0,0.2),
        bA ~ dnorm(0,0.5),
        sigma ~ dexp(1)
    ), data=d )
```

Comparing with PSIS:

```
compare( m1 , m2 , m3 , m4 , m5 , func=PSIS )
```

```
    PSIS    SE dPSIS  dSE pPSIS weight
m1 323.6 16.40   0.0   NA   5.0   0.35
m2 323.7 16.09   0.1 3.63   3.6   0.34
m3 323.9 15.85   0.3 2.92   3.8   0.30
m4 333.6 13.83  10.0 7.26   2.5   0.00
m5 333.8 13.99  10.2 7.28   2.7   0.00
```

So the model with all three predictors is very slightly better than the model with only F and G. To remind you, the DAG from last week is:



The estimates are:

```
precis(m1)
```

```
       mean   sd  5.5% 94.5%
a      0.00 0.08 -0.13  0.13
bF     0.30 0.21 -0.04  0.63
bG    -0.64 0.18 -0.93 -0.35
bA     0.28 0.17  0.01  0.55
sigma  0.93 0.06  0.83  1.03
```

We don't know the true causal effects in this example. The goal is just to use the DAG to reason what these coefficients are estimating, if anything.

First consider F and `bF`. Since G is in the model, the indirect causal effct of F on W is missing. So `bF` only measures the direct path. But it doesn't even do that completely, because A is also in the model. You saw in an earlier lecture that including a cause of the exposure is usually a bad idea, because it statistical reduces variation in the exposure. So `bF` is probably less accurate than if we omitted A. But it estimates the direct causal effect of F on W.

Second consider G. `bG` estimates the direct effect of G on W.

Now what about A? This is a weird one. From the perspective of A, including its mediator F should block all of its association with W. So it isn't a measure of anything, but it is a kind of test of test of the DAG structure. There may be unobserved confounding or more causal paths that explain why A and W remain associated even after stratifying by F. However, since the model without A has almost the same PSIS score as the one with it, maybe there isn't much statistical support for A being associated with W here anyway. A regression that includes only A and F shows no association really between A and W. Why does including G strength the association between A and W? It could just be a fluke of the sample, or it could indicate something is wrong with the causal structure.

3. Start by preparing the data. In this sample, you need to drop the cases with missing values, and that may catch you by surprise.

```
data(cherry_blossoms)
d <- cherry_blossoms
d$D <- standardize(d$doy)
d$T <- standardize(d$temp)
dd <- d[ complete.cases(d$D,d$T) , ]
```

Now I will fit three models and compare them.

```
m3a <- quap(
    alist(
        D ~ dnorm(mu,sigma),
        mu <- a,
        a ~ dnorm(0,10),
        sigma ~ dexp(1)
    ) , data=list(D=dd$D,T=dd$T) )
m3b <- quap(
    alist(
        D ~ dnorm(mu,sigma),
        mu <- a + b*T,
        a ~ dnorm(0,10),
        b ~ dnorm(0,10),
        sigma ~ dexp(1)
    ) , data=list(D=dd$D,T=dd$T) )
m3c <- quap(
    alist(
        D ~ dnorm(mu,sigma),
        mu <- a + b1*T + b2*T^2,
        a ~ dnorm(0,10),
        c(b1,b2) ~ dnorm(0,10),
        sigma ~ dexp(1)
    ) , data=list(D=dd$D,T=dd$T) )
compare( m3a , m3b , m3c , func=PSIS )
```

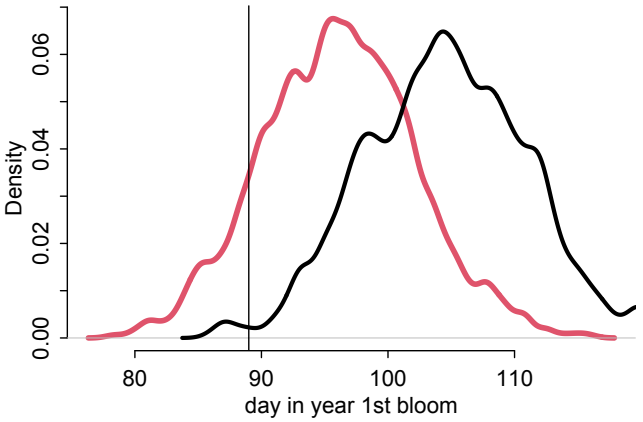
```
      PSIS    SE dPSIS   dSE pPSIS weight
m3b 2112.3 41.00   0.0    NA   2.9   0.71
m3c 2114.1 40.88   1.8  0.26   3.7   0.29
m3a 2199.5 39.57  87.2 16.89   2.1   0.00
```

The linear `m3b` is slightly better than the quadratic `m3c`. Both are much better than the intercept-only `m3a`.

Now we need to generate a predictive distribution for the first day of bloom. We do this by simulating from the model for a specific temperature. The only trick here is to remember that both the predictor and outcome were standardized above. If you didn't standardized, then you won't need to convert back. But my code below does.

```
Tval <- (9 - mean(d$temp,na.rm=TRUE))/sd(d$temp,na.rm=TRUE)
D_sim <- sim( m3b , data=list(T=Tval) )
# put back on natural scale
doy_sim <- D_sim*sd(d$doy,na.rm=TRUE) + mean(d$doy,na.rm=TRUE)
dens( doy_sim , lwd=4 , col=2 , xlab="day in year 1st bloom")
```

```
abline(v=89,lty=1)
dens( d$doy , add=TRUE , lwd=3 )
```



The red density is the predictive distribution for 9 degrees. The black density is the observed historical data. The vertical line is April 1. This is a linear projection, so it is reasonable to question whether such a large degree of continued warming would continue to exert a linear effect on timing of the blossoms. But so far the effect has been quite linear. To do better, we'd need to use some more science, not just statistics.

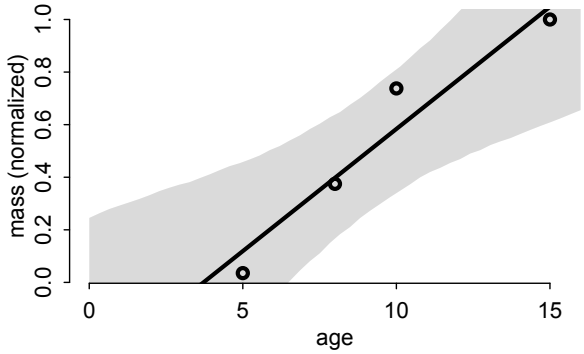
4-OPTIONAL CHALLENGE. I am going to develop this using only one species, then I'll fit to all species. I'll start with the incomparable Apatosaurus.

```
data(Dinosaurs)
d <- Dinosaurs
dd <- d[ d$sp_id==1 , ]
dat <- list(
    A = dd$age,
    M = dd$mass/max(dd$mass)
)
```

I normalized the mass values by the maximum value, so that the maximum is 1 and the other values are a proportion of the maximum. This just makes fitting easier.

Let's start with a respectable straight line. This is a obviously not a biologically sensible model. Animals do not grow at constant rates over their entire lives. But it's a way to start exploring approximations.

```
m4a <- ulam(
    alist(
        M ~ normal(mu,sigma),
        mu <- a + b*A,
        a ~ normal(0,1),
        b ~ normal(0,1),
        sigma ~ exponential(1)
    ) , data=dat , chains=4 , log_lik=TRUE )

plot( dat$A , dat$M , xlab="age" , ylab="mass (normalized)" ,
    lwd=3 , xlim=c(0,16) )
Aseq <- seq(from=0,to=16,len=50)
mu <- link(m4a,data=list(A=Aseq))
lines( Aseq , apply(mu,2,mean) , lwd=3 )
shade( apply(mu,2,PI) , Aseq )
```



Maybe not a bad approximation, but biological nonsense. At ages less than 5, the model predicts negative mass.

Now let's consider something more biologically principled. A classic growth model in biology is the von Bertalanffy growth model. It says that the organism grows at a rate:

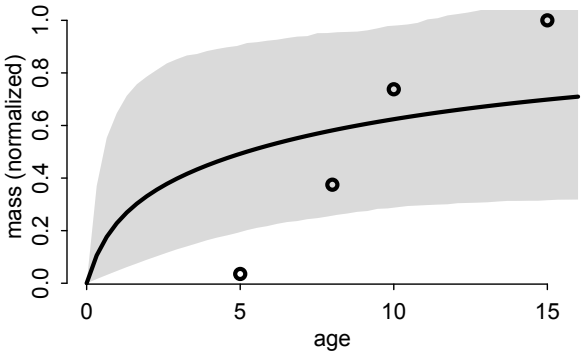
$$\frac{dM}{dA} = b(k - M)$$

where M is mass, A is age, b is a rate and k is the maximum adult size. If we solve the differential equation above, we get:

$$M(A) = k(1 - \exp(-bA))$$

We can use this expression in our statistical model. Like this:

```
m4b <- ulam(
    alist(
        M ~ normal(mu,sigma),
        mu <- k*(1-exp(-b*A)),
        b ~ exponential(1),
        k ~ normal(1,0.5),
        sigma ~ exponential(1)
    ) , data=dat , chains=4 , log_lik=TRUE )

plot( dat$A , dat$M , xlab="age" , ylab="mass (normalized)" ,
    lwd=3 , xlim=c(0,16) )
mu <- link(m4b,data=list(A=Aseq))
lines( Aseq , apply(mu,2,mean) , lwd=3 )
shade( apply(mu,2,PI) , Aseq )
```



This is even worse than the line. The problem is that growth must be slow before age 5 for Apatosaurus. Then it accelerates. So we need some function that can accelerate.

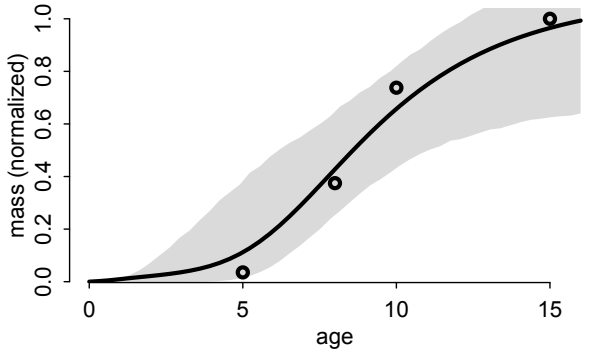
I am going to make our previous growth function accelerate by just raising it to a power. Like this:

$$M(A) = k(1 - \exp(-bA))^a$$

The parameter a is some value above 1 that determines how proportional growth accelerates with age. This is a bit ad hoc. But can you think of a modification to the differential equation that would produce a function like this? Try something that adjusts the leading growth rate b so it isn't constant across age A.

In code form:

```
m4c <- ulam(
    alist(
        M ~ normal(mu,sigma),
        mu <- k*(1-exp(-b*A))^a,
        a ~ exponential(0.1),
        b ~ exponential(1),
        k ~ normal(1,0.5),
        sigma ~ exponential(1)
    ) , data=dat , chains=4 , log_lik=TRUE )

plot( dat$A , dat$M , xlab="age" , ylab="mass (normalized)" ,
    lwd=3 , xlim=c(0,16) )
mu <- link(m4c,data=list(A=Aseq))
lines( Aseq , apply(mu,2,mean) , lwd=3 )
shade( apply(mu,2,PI) , Aseq )
```



Huzzah.

Let's compare these different functions using PSIS. Expect some high Pareto k values.

```
compare( m4a , m4b , m4c , func=PSIS )
```

```
Some Pareto k values are high (>0.5). Set pointwise=TRUE to inspect individual
    PSIS   SE dPSIS  dSE pPSIS weight
m4c -2.8 0.58   0.0   NA   3.5   0.86
m4a  1.1 1.03   3.9 0.71   2.1   0.12
m4b  5.6 1.57   8.4 1.91   1.4   0.01
```

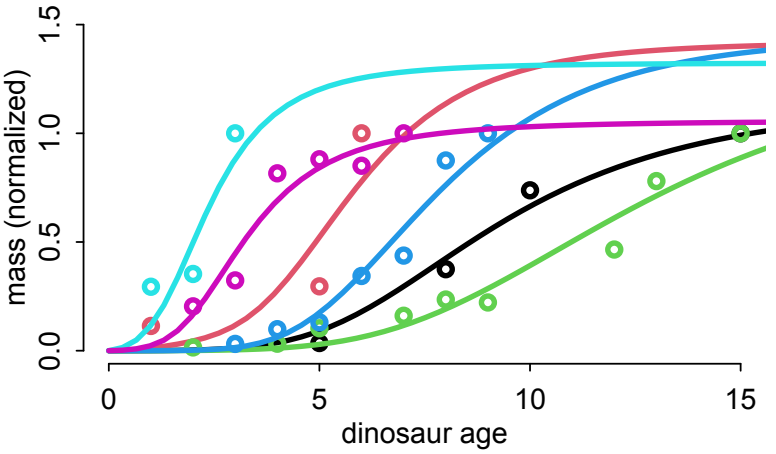
So the last model is best, even considering the standard errors. Still there isn't much data, so don't get excited.

Let's stop fot a moment to talk about model `m4b`. It obviously fails to fit the data. But since it is a biological growth model, its failure is instructive.

It teaches us about what is missing. In this case, it was missing accelerating growth.

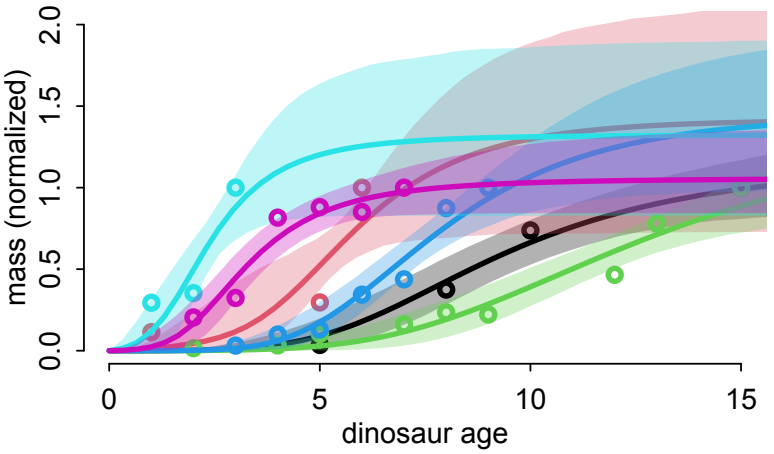
Okay, let's fit all of the dinosaurs now.

```
# scale max size within each species
d$Ms <- sapply( 1:nrow(d) , function(i)
    d$mass[i]/max(d$mass[d$sp_id==d$sp_id[i]]) )
dat_all <- list( A=d$age , M=d$Ms , S=d$sp_id )

m4x <- ulam(
    alist(
        M ~ normal(mu,sigma),
        mu <- k[S]*(1-exp(-b[S]*A))^a[S],
        a[S] ~ exponential(0.1),
        b[S] ~ exponential(1),
        k[S] ~ normal(1,0.5),
        sigma ~ exponential(1)
    ) , data=dat_all , chains=4 , log_lik=TRUE )

plot( NULL , xlim=c(0,max(d$age)) , ylim=c(0,1.5) ,
    xlab="dinosaur age" , ylab="mass (normalized)" )
post <- extract.samples(m4x)
Aseq <- seq(from=0,to=16,len=50)
for ( i in 1:max(d$sp_id) ) {
    j <- which(dat_all$S==i)
    points( dat_all$A[j] , dat_all$M[j] , col=i , lwd=3 )
    mu <- link( m4x , data=list(S=rep(i,50),A=Aseq) )
    lines( Aseq , apply(mu,2,mean) , lwd=3 , col=i )
}
```



There is still a lot of uncertainty for each of these curves. Again with the 89% compatibility intervals of the mean:

```
# with intervals
plot( NULL , xlim=c(0,max(d$age)) , ylim=c(0,2) ,
    xlab="dinosaur age" , ylab="mass (normalized)" )
post <- extract.samples(m4x)
Aseq <- seq(from=0,to=16,len=50)
for ( i in 1:max(d$sp_id) ) {
    j <- which(dat_all$S==i)
    points( dat_all$A[j] , dat_all$M[j] , col=i , lwd=3 )
    mu <- link( m4x , data=list(S=rep(i,50),A=Aseq) )
    lines( Aseq , apply(mu,2,mean) , lwd=3 , col=i )
    shade( apply(mu,2,PI) , Aseq , col=col.alpha(i,0.3) )
}
```



A beautiful mess.

Okay, what have we learned here? In my analysis, the biological model did best, but not the first biological model that I tried. The first biological model did worse than the straight line. But that was helpful, because it pointed out what was missing.