STATISTICAL RETHINKING WINTER 2020
HOMEWORK, WEEK 2 SOLUTIONS

1. The weights that interest us are all adult weights, so we can analyze only the adults and make an okay linear approximation. If you did something else, that's okay. I deliberately made the question a little vague. Loading the data, selecting out adults, and doing the regression from the book:

```
library(rethinking)
data(Howell1)
d <- Howell1
d2 <- d[ d$age >= 18 , ]
xbar <- mean(d2$weight)
m4.3 <- quap(
    alist(
        height ~ dnorm( mu , sigma ) ,
        mu <- a + b*( weight - xbar ) ,
        a ~ dnorm( 178 , 20 ) ,
        b ~ dlnorm( 0 , 1 ) ,
        sigma ~ dunif( 0 , 50 )
    ) , data=d2 )
```

Now we need posterior predictions for each case in the table. Easiest way to do this is to use `sim`. We need `sim`, not just `link`, because we are trying to predict an individual's height. So the relevant compatibility interval includes the Gaussian variance from `sigma`. If you provided only the compatibility interval for μ, that's okay. But be sure you understand the difference.

```
dat <- data.frame( weight=c(45,40,65,31) )
h_sim <- sim( m4.3 , data=dat )
Eh <- apply(h_sim,2,mean)
h_ci <- apply(h_sim,2,PI,prob=0.89)
```

Now all in table form:

```
dat$Eh <- Eh
dat$L89 <- h_ci[1,]
dat$U89 <- h_ci[2,]
round(dat,1)
```

```
  weight    Eh   L89   U89
1     45 154.7 146.7 162.8
2     40 150.4 142.3 158.7
3     65 172.6 163.9 180.9
```

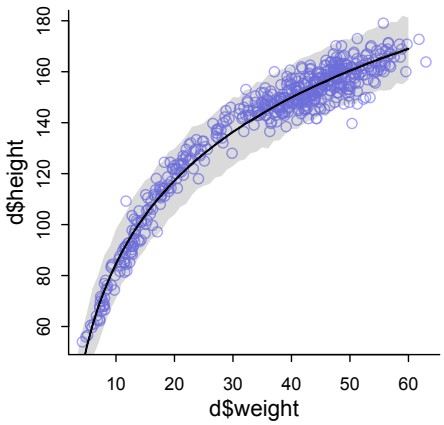
```
4     31 142.0 134.3 150.1
```

2. I left this open to lots of experimentation. But the simplest and probably most effect approach is just a linear regression of height on the log of weight.

```
d$log_weight <- log(d$weight)
xbar <- mean(d$log_weight)
m2 <- quap(
    alist(
        height ~ dnorm( mu , sigma ) ,
        mu <- a + b*( log_weight - xbar ) ,
        a ~ dnorm( 178 , 20 ) ,
        b ~ dlnorm( 0 , 1 ) ,
        sigma ~ dunif( 0 , 50 )
    ) , data=d )

plot( d$weight , d$height , col=col.alpha(rangi2,0.7) )
x_seq <- log(1:60)
mu <- sim( m2 , data=list(log_weight=x_seq) )
mu_mean <- apply(mu,2,mean)
mu_ci <- apply(mu,2,PI,0.99)
lines( exp(x_seq) , mu_mean )
shade( mu_ci , exp(x_seq) )
```

This is what you should see:



You could certainly do better—the trend is under-predicting in the mid ages. But just taking the log of weight does most of the work. Why? It'll help to think of a human body as a cylinder. Roughly. The weight of a cylinder is proportional to its volume. And the volume of a cylinder is:

$$V = \pi r^2 h$$

where r is the radius and h is the height. As the cylinder, uh human, gets taller, the radius gets bigger. So we can just say the radius is some fraction α

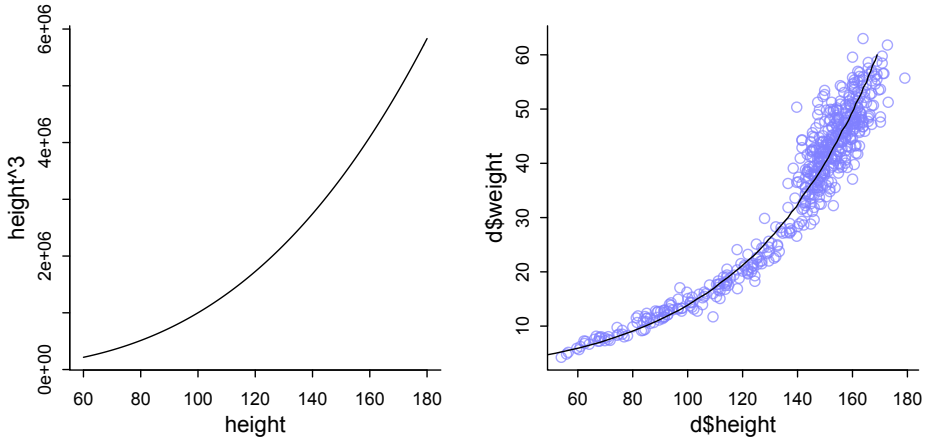
of the height:

$$r = \alpha h$$

Substituting that in:

$$V = \pi \alpha^2 h^3 = k h^3$$

where $k = \pi \alpha^2$ is just some proportionality constant. Now let's plot volume (weight) as a function of height, and I'll compare it to the data viewed the same way:



Not bad. Sometimes physics/biology gets you most of the model. Chapter 16 does this geometric version of the height model in much greater detail.

3. Here is the model, just copied from the chapter:

```
library(rethinking)
data(Howell1)
d <- Howell1
d$weight_s <- ( d$weight - mean(d$weight) )/sd(d$weight)
d$weight_s2 <- d$weight_s^2
m4.5 <- quap(
    alist(
        height ~ dnorm( mu , sigma ) ,
        mu <- a + b1*weight_s + b2*weight_s2 ,
        a ~ dnorm( 178 , 20 ) ,
        b1 ~ dlnorm( 0 , 1 ) ,
        b2 ~ dnorm( 0 , 1 ) ,
        sigma ~ dunif( 0 , 50 )
    ) ,
    data=d )
```

Let's extract the prior:

```
set.seed(45)
prior <- extract.prior( m4.5 )
precis( prior )
```

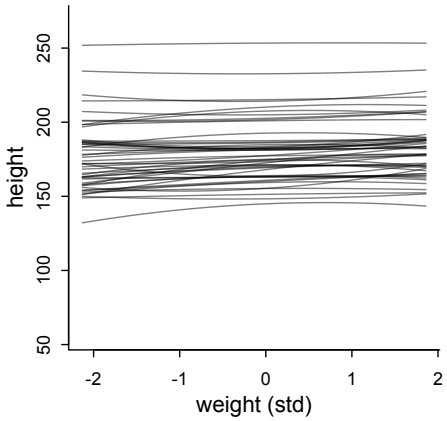
```
'data.frame': 1000 obs. of 4 variables:
        mean    sd   5.5%  94.5%     histogram
a     177.61 20.72 144.18 211.42
b1      1.61  1.88   0.19   4.43
b2     -0.05  0.97  -1.64   1.45
sigma  25.14 14.59   2.52  47.37
```

We want to simulate curves (parabolas) from this prior. One way is to use `link`. Then we won't have to write the linear model again.

```
w_seq <- seq( from=min(d$weight_s) , to=max(d$weight_s) ,
    length.out=50 )
w2_seq <- w_seq^2
mu <- link( m4.5 , post=prior ,
    data=list( weight_s=w_seq , weight_s2=w2_seq ) )
```

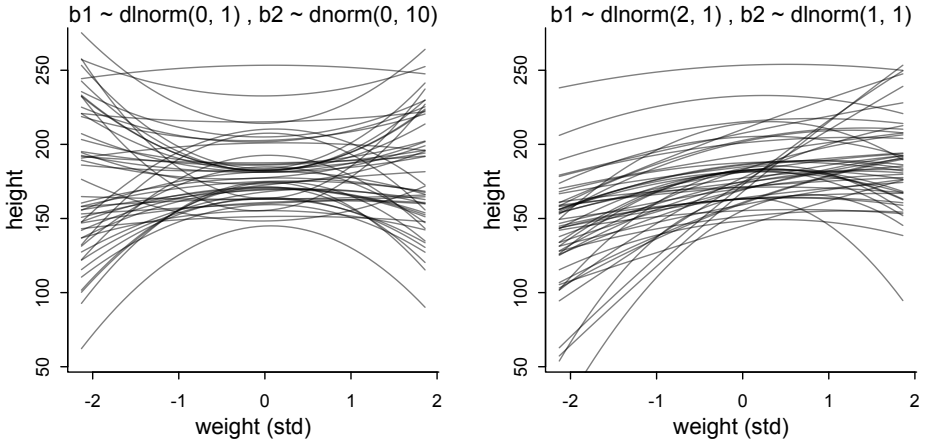
Now `mu` should contain 1000 parabolas. We'll plot just the first 50.

```
plot( NULL , xlim=range(w_seq) , ylim=c(55,270) ,
    xlab="weight (std)" , ylab="height" )
for ( i in 1:50 ) lines( w_seq , mu[i,] , col=col.alpha("black",0.5) )
```

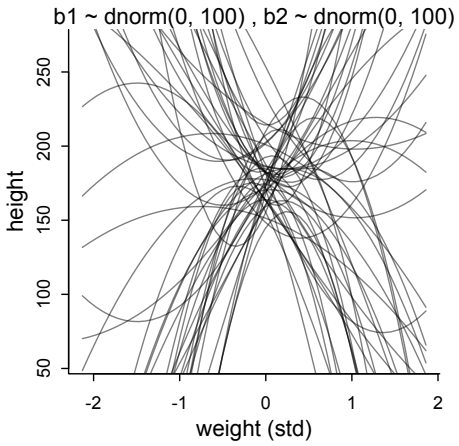


Recall that the world's tallest person was 270cm tall. The tallest person in the sample is about 180cm. The prior curvature is not very strong. Those parabolas hardly bend at all. We can increase the standard deviation on the `b2` prior, but that will produce some silly shapes (left below), where either average weight is tallest or shortest. That can't be right. The basic problem is that `b2` needs to be negative to make the curve bend down, but `b1` has to also change in order to move the maximum height to the right. It's all a bit

confusing, and is they key reason that working with polynomial models is so hard. The prior on the right below can only bend down, but I've made the linear model `a + b1*weight_s - b2*weight_s2` and given `b2` a log-Normal prior.



A key problem in getting reasonable curves here is that obviously `a` and `b1` and `b2` are correlated in the family of reasonable curves. But the priors are uncorrelated—they are independent of one another. Still, if you can get independent priors to at least live within some reasonably space of outcome values, that's a lot better than flat priors. What would flat priors look like here? Something like this:



These prior curves actually strongly favor explosive growth or shrinkage near the mean. This is a general phenomenon with "flat" priors: Once the predictor is at all complicated, "flat" does not imply "no relationship."

Do any of the priors above make a difference for inference in this sample? No. There is a lot of data and the model is quite simple, in terms of the way that parameters relate to predictions. This will not always be the case.