Statistical Rethinking Exercise - Chapter 4

Yue - Mar.4, 2018

chapter 4

Easy - model definition

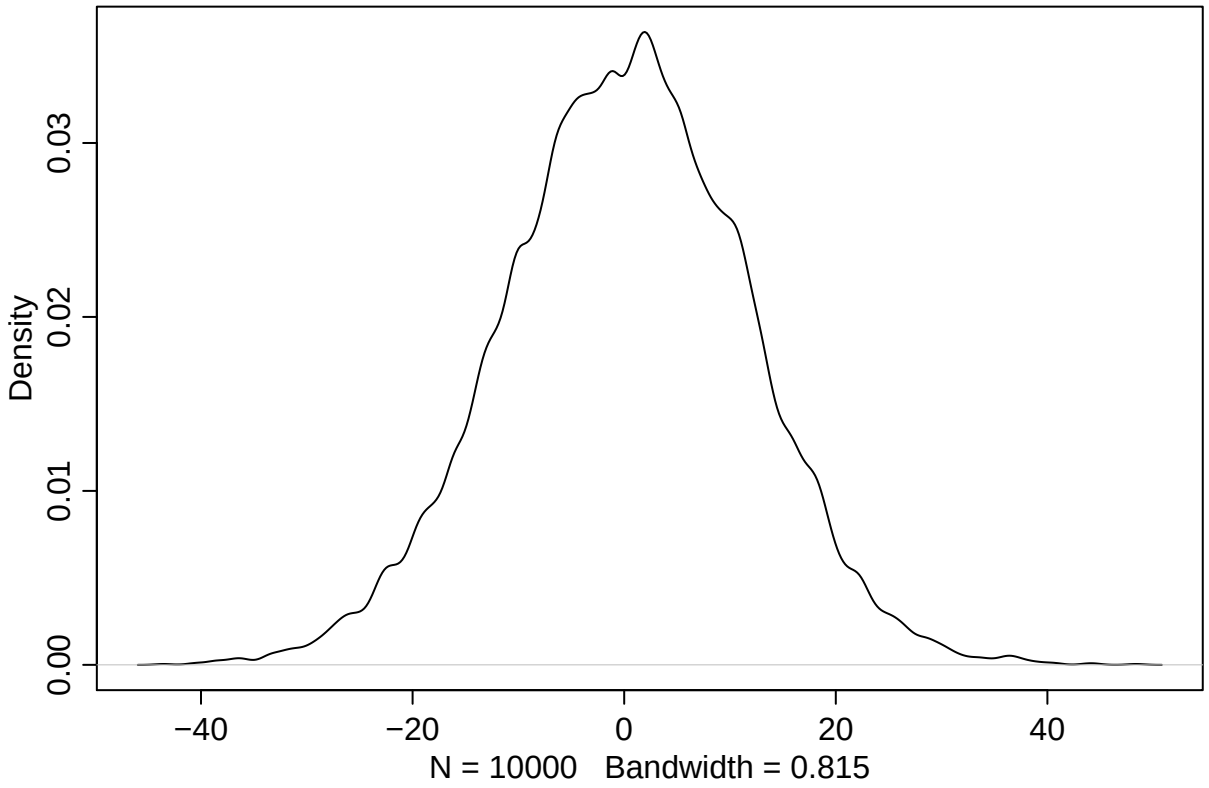
y_i $Normal(\mu, \sigma)$ – likelihood
μ $Normal(0, 10)$ – μ prior
σ $Uniform(0, 10)$ – σ prior
2 parameters are in the posterior distribution
Form of Bayes' theorem that includes the proper likelihood and priors.

y_i $Normal(\mu, \sigma)$ – likelihood
$\mu_i = \alpha + \beta x_i$ – linear model
α $Normal(0, 10)$ – α prior
β $Normal(0, 1)$ – β prior
σ $Uniform(0, 10)$ – σ prior
3 parameters are in the posterior distribution

Medium

4M1 Simulate observed heights from the prior
y_i $Normal(\mu, \sigma)$ – likelihood
μ $Normal(0, 10)$ – μ prior
σ $Uniform(0, 10)$ – σ prior

```
sample_mu <- rnorm( 1e4 , 0 , 10 )
sample_sigma <- runif( 1e4 , 0 , 10 )
sim_prior <- rnorm( 1e4 , sample_mu , sample_sigma )
dens( sim_prior )
```



4M2 Translate the model into map formula
flist<- alist(y ~ dnorm(mu , sigma) , mu ~ dnorm(0 , 10) , sigma ~ dunif(0 , 10))

4M3 Translate the map formula into a mathematical model definition
flist<- alist(y ~ dnorm(mu , sigma) , mu <- a + b*x, a ~ dnorm(0 , 50) , b ~ dunif(0 , 10) , sigma ~ dunif(0 , 50))
y_i $Normal(\mu_i, \sigma)$ – likelihood
$\mu_i = \alpha + \beta x_i$ – linear model
α $Normal(0, 50)$ – α prior
β $Uniform(0, 10)$ – β prior
σ $Uniform(0, 50)$ – σ prior

4M4 height model
height.model <- map(alist(height ~ dnorm(mu , sigma) , mu <- a + b*year, a ~ dnorm(115 , 8) , #6-yr olds, white boys b ~ dnorm(5 , 2) , sigma ~ dunif(0 , 5)) , data=sample)

4M5 If average height in first year=120cm, student got taller each year. Does this information lead you to change your choice of priors? Why?
set: a ~ dnorm(120 , 8)

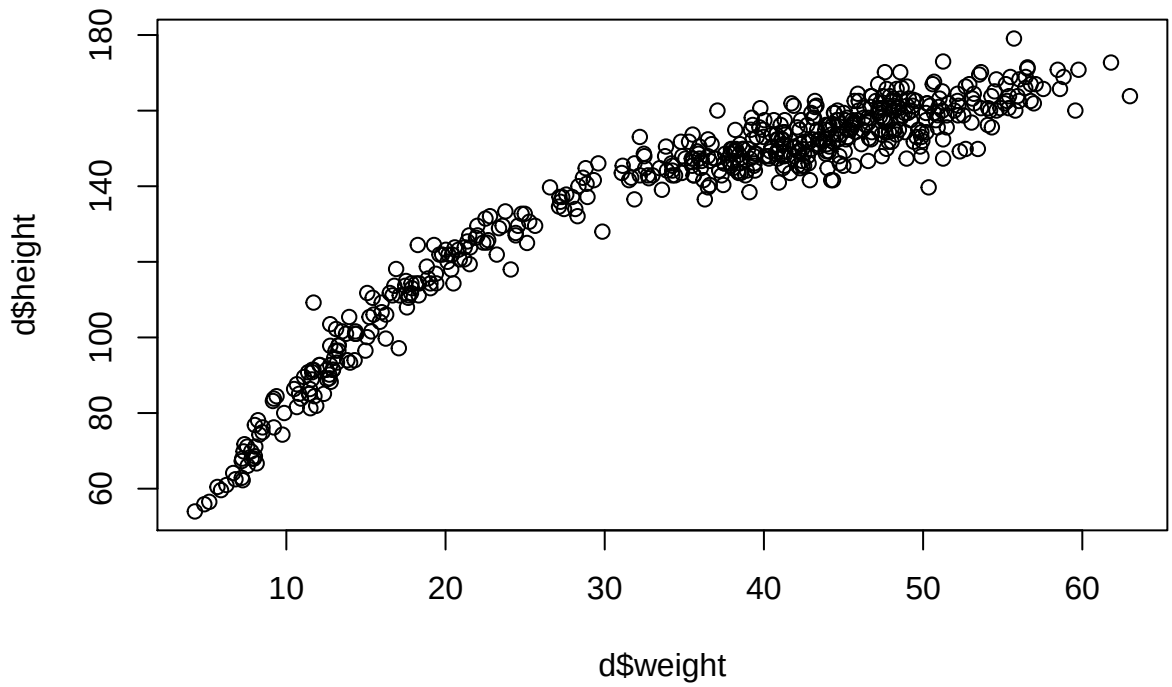
4M6 variance among heights for students of the same age is never more than 64 cm, how does this

lead you to revise your priors?
set: sigma ~ dunif(0 , 64) ?

Hard

4H1 predict height with 89% intervals for individuals given weight

```
# get height data
data(Howell1)
d <- Howell1

#plot raw data
plot(d$height~d$weight)
```




```
# build model using map
m1 <- map(
  alist(
    height ~ dnorm (mu, sigma),
    mu <- a + b * weight,
    a ~ dnorm(165, 80),
```

```
    b ~ dnorm(0,10),
    sigma ~ dunif(0,64) #given information in 4M6
  ),
  data = d
)

# simulate height based on model
sample.weight <-as.data.frame(cbind(c(1:5),c(46.95,43.72,64.78,32.59,54.63)))
colnames(sample.weight)<-c("id","weight")

# use link to compute mu for each sample from posterior and for each weight
mu <- link( m1 , data=data.frame(weight=sample.weight$weight))
```

```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```
# summarize the distribution of mu
mu.mean <- apply( mu , 2 , mean )
mu.HPDI <- apply( mu , 2 , HPDI , prob=0.89 )
mu.PI <- apply( mu , 2 , PI , prob=0.89 )

# simulate height
sim.height <- sim( m1 , data=list(weight=sample.weight$weight) , n=1e4 )
```

```
## [ 1000 / 10000 ]
[ 2000 / 10000 ]
[ 3000 / 10000 ]
[ 4000 / 10000 ]
[ 5000 / 10000 ]
[ 6000 / 10000 ]
[ 7000 / 10000 ]
[ 8000 / 10000 ]
```

```
[ 9000 / 10000 ]
[ 10000 / 10000 ]
```

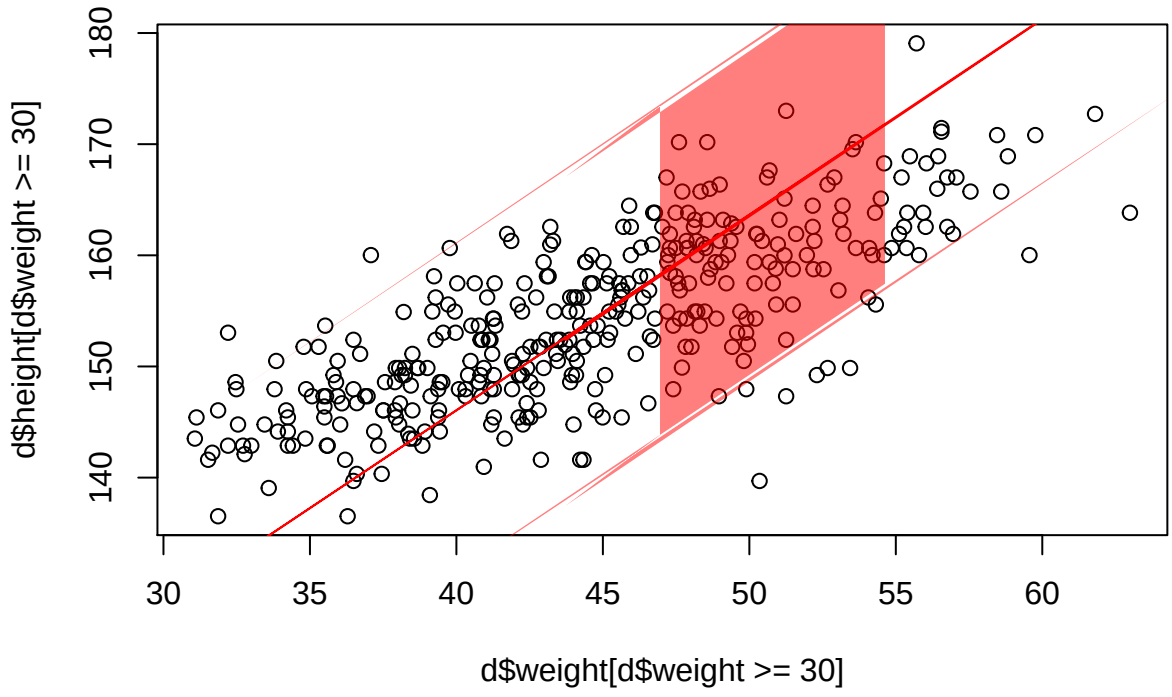
```
# summarize the distribution of simulation
sim.mean <- apply( sim.height , 2 , mean )
sim.HPDI <- apply( sim.height , 2 , HPDI , prob=0.89 )
sim.PI <- apply( sim.height , 2 , PI , prob=0.89 )

cbind(sim.mean, mu.mean)
```

```
##      sim.mean  mu.mean
## [1,] 158.2470 158.2833
## [2,] 152.3824 152.5857
## [3,] 189.6799 189.7350
## [4,] 133.0176 132.9526
## [5,] 171.7877 171.8306
```

```
cbind(sim.HPDI, mu.HPDI, sim.PI, mu.PI) #almost the same
```

```
##           [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]
## |0.89 143.8244 137.4619 174.9373 118.4209 157.4381 157.4494 151.8473
## 0.89| 173.4263 167.0490 205.0314 147.9168 187.2994 158.9963 153.2764
##           [,8]     [,9]    [,10]    [,11]    [,12]    [,13]    [,14]
## |0.89 188.4609 132.3268 170.7754 143.4754 137.6281 174.7198 118.3225
## 0.89| 191.0535 133.6276 172.7330 173.1162 167.3300 204.8552 147.8520
##          [,15]    [,16]    [,17]    [,18]    [,19]    [,20]
## |0.89 156.8687 157.5151 151.8771 188.4076 132.3151 170.8401
## 0.89| 186.8321 159.0776 153.3281 191.0400 133.6232 172.8179
```

```
#plot adding MAP line and HPDI
plot(d$height[d$weight>=30]~d$weight[d$weight>=30])
lines( sample.weight$weight, sim.mean , col="red") # draw MAP line
shade( sim.HPDI , sample.weight$weight, col=col.alpha("red",0.5) ) # draw HPDI region for line
```




```
#shade( sim.PI , sample.weight$weight, col=col.alpha("green",0.3) ) # draw PI region for simul

# check 1st individual
posterior <- extract.samples(m1)
sim1 <- rnorm(n = 1e4, mean = posterior$a + posterior$b*sample.weight$weight[1], sd = posterio
mean(sim1)
```

```
## [1] 158.1683
```

```
PI(samples = sim1, prob = .89)
```

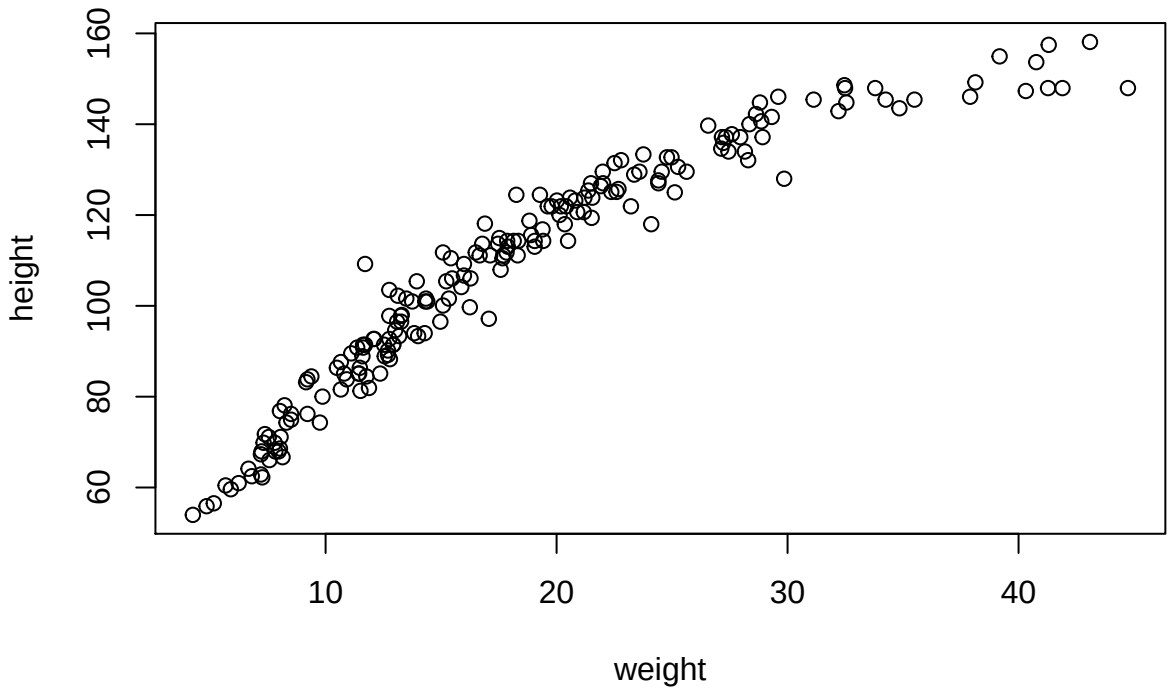
```
##       5%      94%
## 143.2636 173.0091
```

4H2 below 18yr sample

```
# get below 18 year subset
d2 <- Howell1[Howell1$age<18,]
dim(d2)
```

```
## [1] 192   4
```

```
plot(height~weight, data=d2)
```



(a) Linear regression using map.

```
m2 <- map(
        alist(
            height ~ dnorm( mu , sigma ) ,
            mu <- a + b*weight,
            a ~ dnorm( 110 , 80 ) ,
            b ~ dnorm( 0 , 10 ) ,
            sigma ~ dunif( 0 , 64 )
        ) ,
        data=d2 )
m2
```

```
##
## Maximum a posteriori (MAP) model fit
##
## Formula:
## height ~ dnorm(mu, sigma)
```

```
## mu <- a + b * weight
## a ~ dnorm(110, 80)
## b ~ dnorm(0, 10)
## sigma ~ dunif(0, 64)
##
## MAP values:
##         a         b     sigma
## 58.248237  2.719290  8.437109
##
## Log-likelihood: -681.9
```

```
precis(m2, prob=0.95)
```

```
##         Mean StdDev  2.5% 97.5%
## a     58.25   1.40 55.51 60.99
## b      2.72   0.07  2.59  2.85
## sigma  8.44   0.43  7.59  9.28
```

```
post<-extract.samples(m2)
mean(post$b)*10 # 10 units of increase in weight
```

```
## [1] 27.19767
```

(b) Plot the raw data height ~ weight (see above).
Superimpose the MAP regression line and 89% HPDI for the mean.

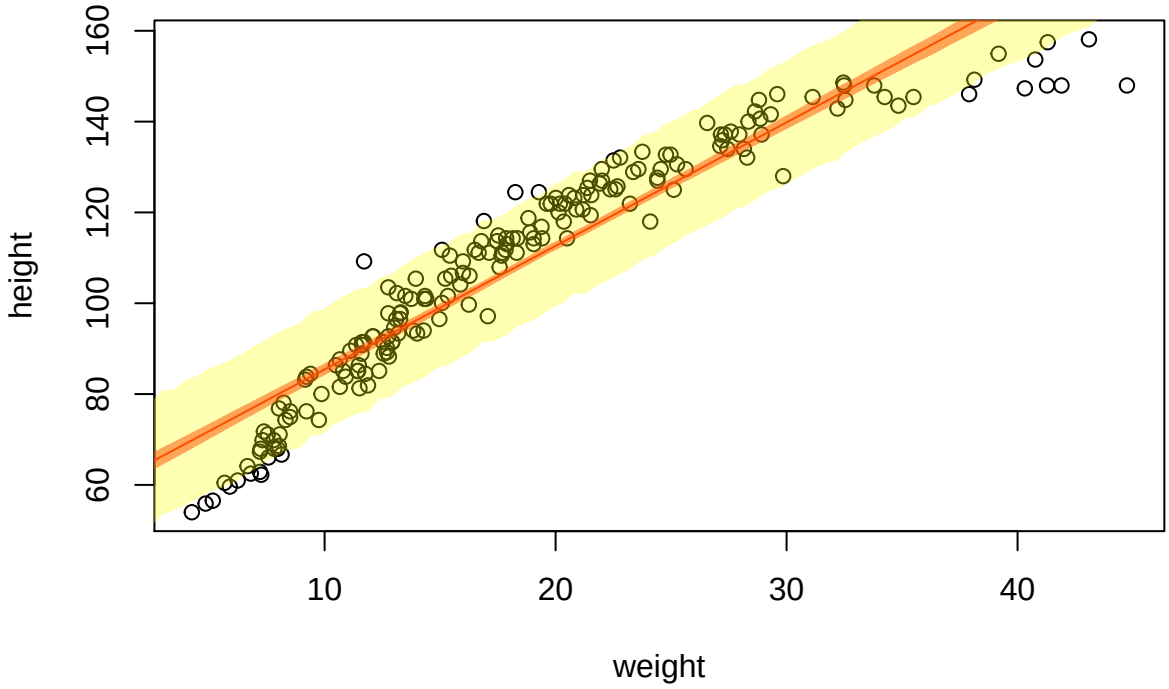
```
# simulate mu
  weight.seq <- seq(from = 1, to = 50, length.out = 100)

  #method 1 using link function
  mu <- link( m2 , data=data.frame(weight=weight.seq) )
```

```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```r
  mu.mean <- apply( mu , 2 , mean )
  #method 2 define linear model
  mu.link <- function(weight) post$a + post$b*weight
  mu2 <- sapply( weight.seq , mu.link )
  mu2.mean <- apply( mu2 , 2 , mean ) #almost the same

  #calculate PI
  mu.HPDI <- apply( mu , 2 , HPDI , prob=0.89 )
  mu.PI <- apply( mu , 2 , PI , prob=0.89 )

# predict height
  sim.height <- sim( m2 , data=list(weight=weight.seq) , n=1e4 )
```

```
## [ 1000 / 10000 ]
[ 2000 / 10000 ]
[ 3000 / 10000 ]
[ 4000 / 10000 ]
[ 5000 / 10000 ]
[ 6000 / 10000 ]
[ 7000 / 10000 ]
[ 8000 / 10000 ]
[ 9000 / 10000 ]
[ 10000 / 10000 ]
```

```r
  # summarize the distribution of simulation
  sim.mean <- apply( sim.height , 2 , mean )
  sim.HPDI <- apply( sim.height , 2 , HPDI , prob=0.89 )
  sim.PI <- apply( sim.height , 2 , PI , prob=0.89 )

#plot adding MAP line and HPDI
  plot(height~weight, data=d2)
  lines( weight.seq, mu.mean , col="red") # draw MAP line
  shade( mu.HPDI , weight.seq, col=col.alpha("red",0.5) ) # draw HPDI region for mean
  shade( sim.HPDI , weight.seq, col=col.alpha("yellow",0.3) ) # draw HPDI region for predicted
```



Linear model height~weight does not seem a good fit given the data and graph above. Try linear model with transformation.

4H3 log weight as the predictor

(a) fit model

h_i $Normal(\mu_i, \sigma)$ – likelihood

$\mu_i = \alpha + \beta log(w_i)$ – linear model

α $Normal(178, 100)$ – α prior

β $Normal(0, 100)$ – β prior

σ $Uniform(0, 50)$ – σ prior

```
m3 <- map(
        alist(
            height ~ dnorm( mu , sigma ) ,
            mu <- a + b*log(weight),
            a ~ dnorm( 178 , 100 ) ,
            b ~ dnorm( 0 , 100 ) ,
            sigma ~ dunif( 0 , 50 )
        ) ,
        data=d )
```

```
m3
```

```
##
## Maximum a posteriori (MAP) model fit
##
## Formula:
## height ~ dnorm(mu, sigma)
## mu <- a + b * log(weight)
## a ~ dnorm(178, 100)
## b ~ dnorm(0, 100)
## sigma ~ dunif(0, 50)
##
## MAP values:
##          a          b      sigma
## -23.784882  47.075692   5.134965
##
## Log-likelihood: -1661.9
```

```
precis(m3, prob=0.95)
```

```
##         Mean StdDev   2.5%  97.5%
## a     -23.78   1.34 -26.40 -21.17
## b      47.08   0.38  46.33  47.83
## sigma   5.13   0.16   4.83   5.44
```

```
invisible(post3 <- extract.samples(m3))
```

model: height = -23.9257382, -23.7037561, -25.2007224, -23.3071135, -24.7846083, -22.5985707, -23.8586346, -23.5555592, -24.7348599, -22.07731, -22.5108478, -25.5073056, -23.0210515, -23.7641274, -24.6589264, -22.3860537, -24.6583915, -23.2076833, -23.5573083, -21.6977349, -23.1787273, -21.323631, -22.6476044, -23.289734, -23.7396277, -21.7771509, -23.9031218, -22.1498691, -24.9147646, -24.2544507, -23.1573395, -26.9368598, -22.9628983, -25.2156804, -21.6279785, -25.9611505, -24.3340551, -21.62937, -25.0451945, -24.7279831, -23.3087664, -25.9984772, -24.0984365, -23.6618074, -22.3285542, -24.3405747, -23.1768605, -22.6647748, -22.5110506, -22.8810813, -24.55954, -26.0787208, -22.3070671, -23.7643971, -23.1857282, -23.0521392, -22.5341392, -23.3184576, -23.94156, -22.2117171, -21.9678663, -23.455654, -23.3159535, -25.0257611, -25.0283409, -23.7876183, -22.9570647, -24.9495627, -22.8384747, -22.1441754, -23.1443096, -24.042638, -23.9093469, -22.8853935, -22.7577512, -23.4972884, -23.7344661, -24.2319627, -26.9201402, -24.1291003, -23.0926719, -22.688034, -24.2752489, -24.8193076, -23.2873116, -26.5536722, -24.5165072, -24.7011072, -22.9245972, -25.3158935, -23.5963342, -24.7215704, -21.6136329, -24.0740587, -23.8691209, -24.757832, -24.1186719, -24.813285, -24.8286878, -23.6634934, -23.5725319, -

47.1489344, 47.3721273, 47.0688114, 46.9182421, 47.0196909, 47.5163481, 47.2717018, 46.9434664, 46.8416013, 46.6914047, 46.7297567, 47.0718143, 46.5726997, 47.5236592, 47.31119, 47.2250663, 46.4581114, 47.6831243, 47.369304, 46.5492469, 46.9922389, 46.8464423, 47.0320974, 46.8710676, 47.5732885, 47.2474006, 46.2734932, 47.3806492, 47.041723, 47.2384606, 46.9602324, 47.1008364, 47.0170534, 46.6617655, 46.7970258, 47.1916739, 47.1036092, 47.1803003, 46.9851149, 47.3278388, 47.7310326, 47.6601811, 46.8296022, 47.0854108, 47.8238193, 46.7960932, 47.1740927, 47.5201462, 47.2895485, 47.1132366, 47.583646, 46.9051437, 46.3004, 47.0303958, 47.7954351, 47.1687203, 47.2917327, 46.667752, 46.966018, 47.2710208, 47.493108, 47.1738891, 47.3774739, 47.0812933, 47.0167942, 47.2048915, 46.7047012, 46.3038596, 46.8787942, 46.9835806, 47.4853494, 47.2377397, 47.0360823, 47.2041551, 47.3352298, 46.7550789, 46.4336878, 47.0821211, 46.8776931, 47.5144762, 47.5724185, 46.83184, 47.0711634, 47.7428107, 46.7917746, 46.5086515, 46.9900354, 47.2314979, 47.0804114, 46.9017979, 46.6331518, 46.4216618, 47.2770604, 47.4976669, 46.6479227, 47.0220994, 46.5064482, 46.5958458, 47.2113958 * log(weight)

(b) plot: use samples from the quadratic approximate posterior of the model in (a)

1. superimpose the predicted mean height as a function of weight

2. the 97% HPDI for the mean

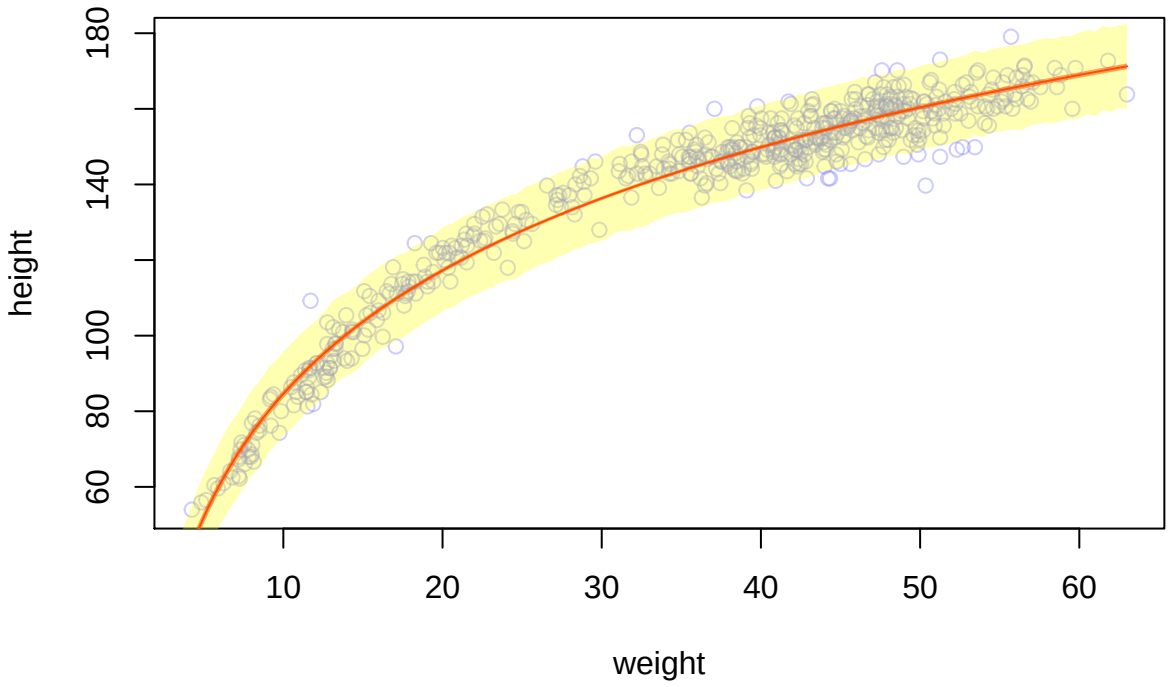
3. the 97% HPDI for predicted height

```
# simulate mu
  weight.seq <- seq(from = 1, to = max(d$weight), length.out = ((max(d$weight)-1)*2) )

  mu3 <- sapply(weight.seq, function(weight) post3$a + post3$b * log(weight) ) #model
  mu3.mean <- apply( mu3 , 2 , mean )
  mu3.HPDI <- apply( mu3 , 2 , HPDI , prob=0.97 )   #97% HDPI of the mean

# predict height
  sim3 <- sim( m3 , data=list(weight=weight.seq) , n=1e4 )
```

```
## [ 1000 / 10000 ]
[ 2000 / 10000 ]
[ 3000 / 10000 ]
[ 4000 / 10000 ]
[ 5000 / 10000 ]
[ 6000 / 10000 ]
[ 7000 / 10000 ]
[ 8000 / 10000 ]
[ 9000 / 10000 ]
```

```
[ 10000 / 10000 ]
```

```
  sim3.HPDI <- apply( sim3 , 2 , HPDI , prob=0.97 )   #97% HDPI of predicted height

#plot adding MAP line and HPDI
  plot( height~weight, data=d, col=col.alpha(rangi2, 0.4)) #R code 4.73
  lines( weight.seq, mu3.mean , col="red") # draw MAP line
  shade( mu3.HPDI , weight.seq, col=col.alpha("red",0.5) ) # draw HPDI region for mean
  shade( sim3.HPDI , weight.seq, col=col.alpha("yellow",0.3) ) # draw HPDI region for predicte
```



The height ~ log(weight) seems a much better fit comparing to the height~weight model.