hw6 for stat341

Zhihong Zhang

Feb 20th,2017

Q: 4H1-4H3

4H1.The weights listed below were recorded in the !Kung census, but heights were not recorded for these individuals. Provide predicted heights and 89% intervals (either HPDI or PI) for each of these individuals. That is, fill in the table below, using model-based predictions.

Individual weight expected height 89% interval 1 46.95
2 43.72
3 64.78
4 32.59
5 54.63

Solution:

assume the weight is for adult

```
set.seed(112)
Kung<- (d2)
kunglist <- alist(
height ~ dnorm( mu , sigma ),
mu <- a + b*weight,
a ~ dnorm( 178, 100 ),
b ~ dunif( 0 , 100 ),
sigma ~ dunif( 0 , 50 )
)
modelKung<-  map(kunglist, data = Kung)
coef(modelKung)
```

```
##           a           b       sigma
## 113.9024670   0.9045266   5.0718716
```

```
post <- extract.samples(modelKung, n = 1e4)
```

for weight is 46.95.

```
mu_at_46.95 <- post$a + post$b * 46.95
mean(mu_at_46.95) #expected height
```

```
## [1] 156.3717
```

```
HPDI(mu_at_46.95, prob = 0.89)
```

```
##    |0.89    0.89|
## 155.9106 156.8096
```

for weight is 43.72.

```
mu_at_43.72 <- post$a + post$b * 43.72
mean(mu_at_43.72) #expected height
```

```
## [1] 153.4499
```

```r
HPDI(mu_at_43.72, prob = 0.89)
```

```
##    |0.89    0.89|
## 153.0077 153.8887
```

for weight is 64.78.

```r
mu_at_64.78 <- post$a + post$b * 64.78
mean(mu_at_64.78) #expected height
```

```
## [1] 172.5004
```

```r
HPDI(mu_at_64.78, prob = 0.89)
```

```
##    |0.89    0.89|
## 171.0489 173.8428
```

for weight is 32.59.

```r
mu_at_32.59 <- post$a + post$b * 32.59
mean(mu_at_32.59) #expected height
```

```
## [1] 143.3819
```

```r
HPDI(mu_at_32.59, prob = 0.89)
```

```
##    |0.89    0.89|
## 142.4607 144.3320
```

for weight is 54.63.

```r
mu_at_54.63 <- post$a + post$b * 54.63
mean(mu_at_54.63) #expected height
```

```
## [1] 163.3189
```

```r
HPDI(mu_at_54.63, prob = 0.89)
```

```
##    |0.89    0.89|
## 162.5576 164.1121
```

4H2. Select out all the rows in the Howell1 data with ages below 18 years of age. If you do it right, you should end up with a new data frame with 192 rows in it. (a) Fit a linear regression to these data, using map. Present and interpret the estimates. For every 10 units of increase in weight, how much taller does the model predict a child gets? (b) Plot the raw data, with height on the vertical axis and weight on the horizontal axis. Superimpose the MAP regression line and 89% HPDI for the mean. Also superimpose the 89% HPDI for predicted heights. (c) What aspects of the model fit concern you? Describe the kinds of assumptions you would change, if any, to improve the model. You don't have to write any new code. Just explain what the model appears to be doing a bad job of, and what you hypothesize would be a better model.

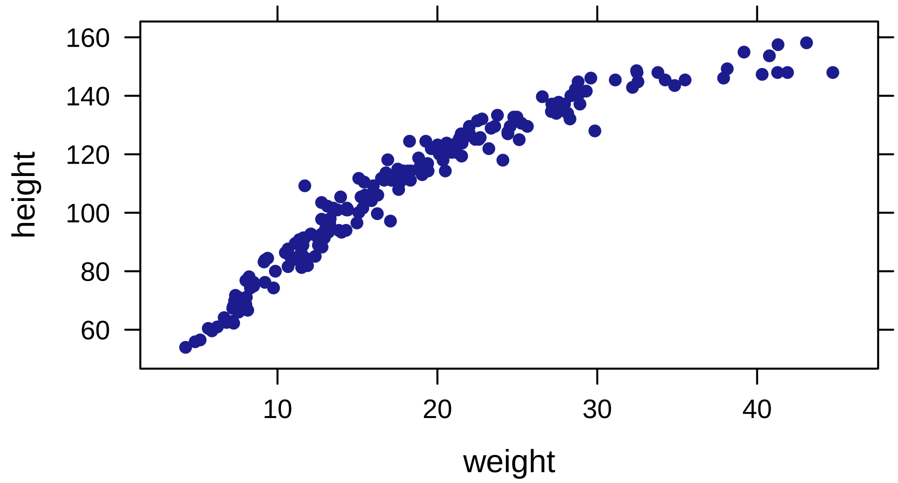
Solution:

(a)

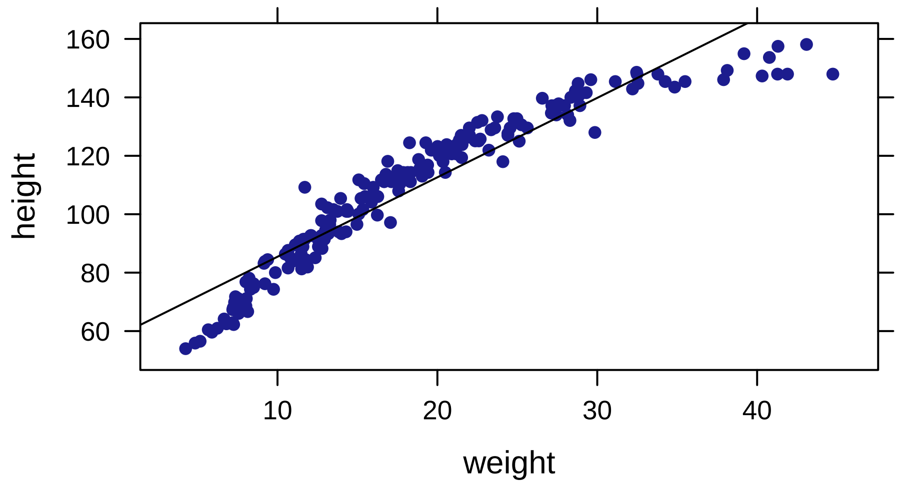
For every 10 units of increase in weight, About 30 cm taller does the model predict a child gets

```r
set.seed(14)
data(Howell1)
HowellnonAdults <- Howell1 %>% filter(age < 18) #Howell1 data with ages below 18 years of age

flist <- alist(
height ~ dnorm( mu , sigma ),
```

```
mu <- a + b*weight,
a ~ dnorm( 156, 100 ),
b ~ dunif( 0 , 10 ),
sigma ~ dunif( 0, 50 )
)

linearmodela <- map(flist, data = HowellnonAdults)
xyplot(height ~ weight, data = HowellnonAdults)
```




```
plotFun(a + b * x ~ x, a = coef(linearmodela)["a"], b = coef(linearmodela)["b"], add = TRUE, col = "bla
```



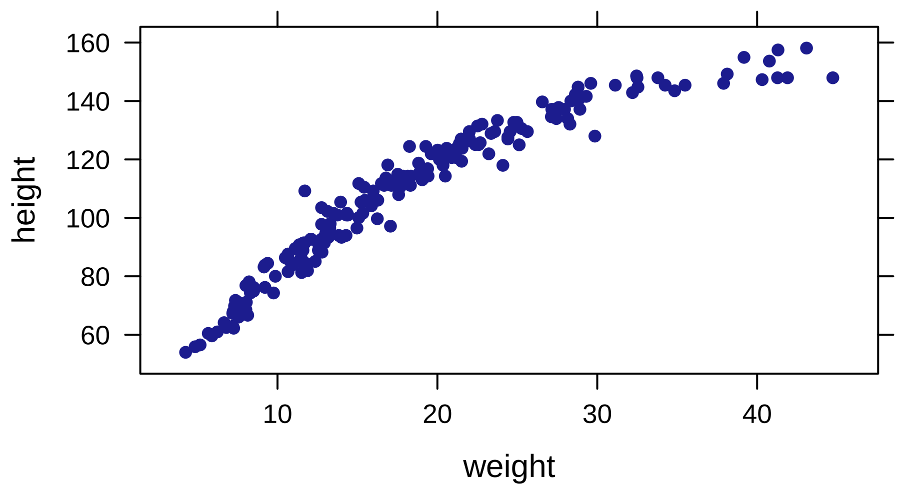
(b)

```
set.seed(124)
data(Howell1)
HowellnonAdults <- Howell1 %>% filter(age < 18) #Howell1 data with ages below 18 years of age
```

```
flist <- alist(
height ~ dnorm( mu , sigma),
mu <- a + b*weight,
a ~ dnorm( 156, 100 ),
b ~ dunif( 0 , 10 ),
sigma ~ dunif(0, 50 )
)

linearmodelb <- map(flist, data = HowellnonAdults)
precis(linearmodelb)
```

```
##        Mean StdDev  5.5% 94.5%
## a     58.25   1.40 56.02 60.48
## b      2.72   0.07  2.61  2.83
## sigma  8.44   0.43  7.75  9.12
```

```
xyplot(height ~ weight, data = HowellnonAdults)
```



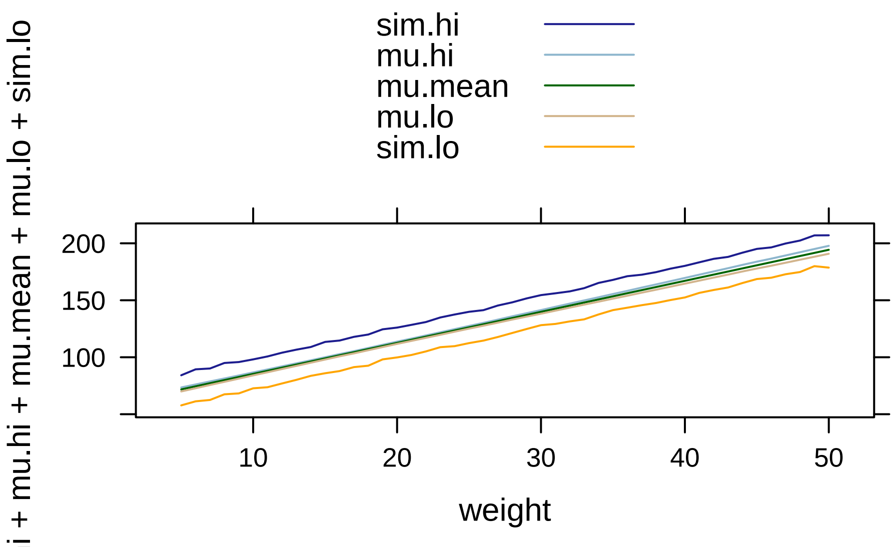

```
linearmodelb.pred <-
  data_frame(
    weight = seq(from = 5, to = 50, by = 1)
  )

mu <- link(linearmodelb, data = linearmodelb.pred)
```

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

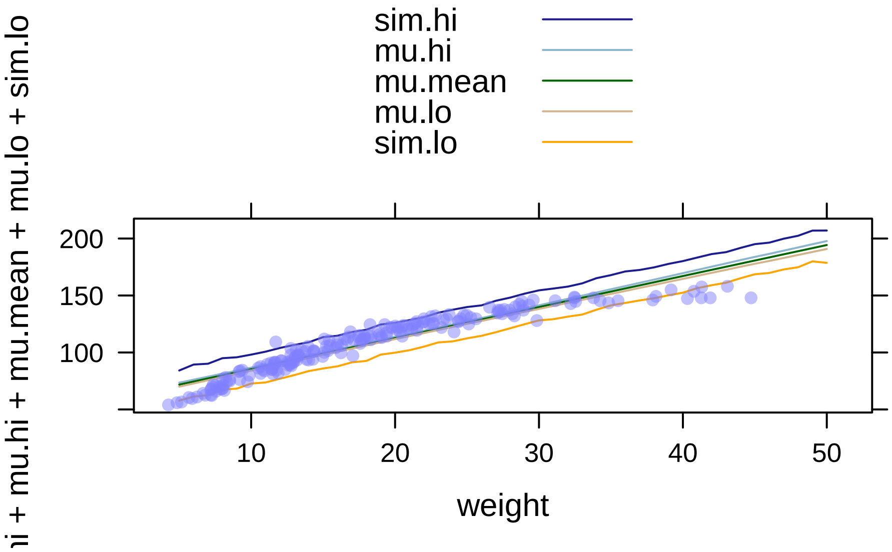
```
sim.height <- sim(linearmodelb, data = linearmodelb.pred)
```

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

```
linearmodelb.pred <-
  linearmodelb.pred %>%
  mutate(
    mu.mean = apply(mu, 2, mean),
    mu.lo = apply(mu, 2, HPDI,prob=0.89)[1,],
    mu.hi = apply(mu, 2, HPDI,prob=0.89)[2,],
    sim.lo = apply(sim.height, 2, HPDI,prob=0.89)[1,],
    sim.hi = apply(sim.height, 2, HPDI,prob=0.89)[2,]
    )
xyplot(sim.hi + mu.hi + mu.mean + mu.lo + sim.lo ~ weight,
       data = linearmodelb.pred,  type = "l", auto.key = list(lines = TRUE, points = FALSE))
```




```
plotPoints(height ~ weight, data = HowellnonAdults, col = rangi2, alpha = 0.5, add = TRUE)
```



(c) The linear fit does not perfectly fit the curve and such things. I would change the linear model to improve the improve the performance. I think the quadratic or third order polynomial may do a better job on curve.

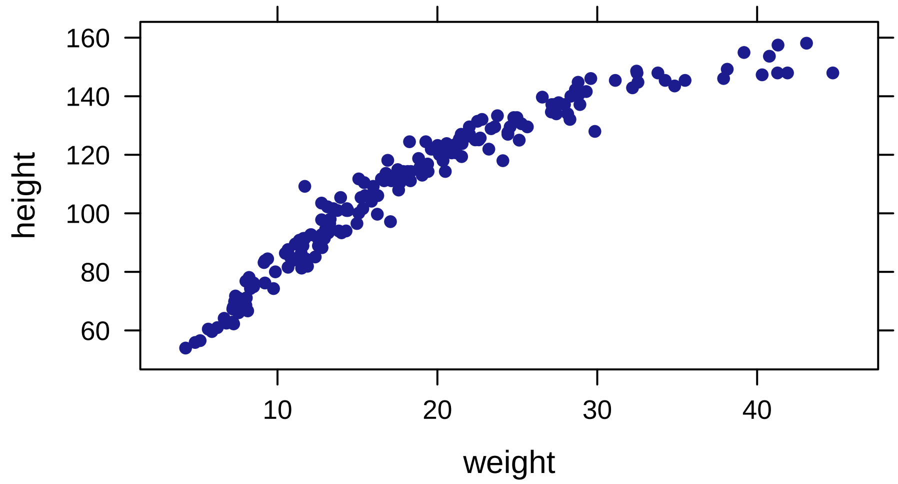
4H3. Suppose a colleague of yours, who works on allometry, glances at the practice problems just above. Your colleague exclaims, "That's silly. Everyone knows that it's only the logarithm of body weight that scales with height!" Let's take your colleague's advice and see what happens. (a) Model the relationship between height (cm) and the natural logarithm of weight (log-kg). Use the entire Howell1 data frame, all 544 rows, adults and non-adults. Fit this model, using quadratic approximation:

$y_i \sim Normal(\mu, \sigma)$
$\mu_i = \alpha + \beta \cdot log(w_i)$
$\alpha \sim Normal(178, 100)$
$\beta \sim Normal(0, 100)$
$\sigma \sim Uniform(0, 50)$

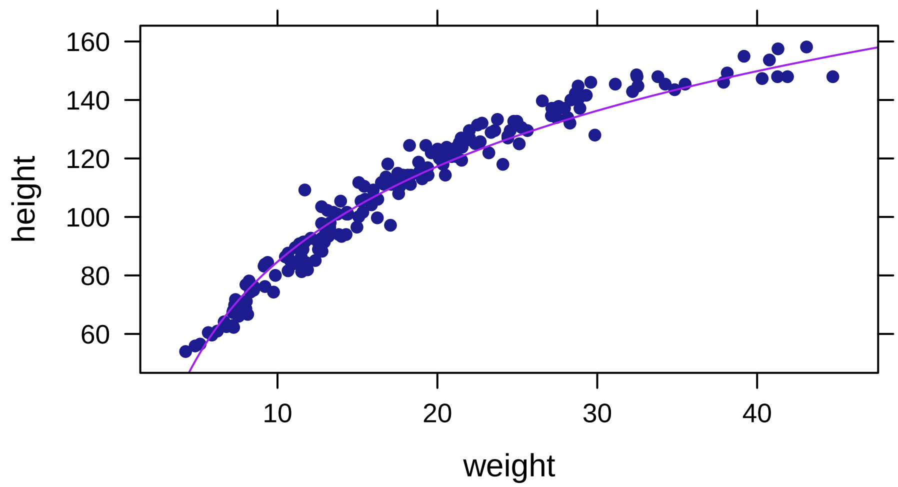
where hi is the height of individual i and wi is the weight (in kg) of individual i. The function for computing a natural log in R is just log. Can you interpret the resulting estimates?

Solution:

```
set.seed(134)



linearmodel <- map(
  alist(
    height ~ dnorm(mu, sigma),
    mu <- a + b * log(weight) ,
    a ~ dnorm(178, 100),
    b ~ dnorm(0, 10),
    sigma ~ dunif(0, 50)
  ),
  data = Howell1
)

xyplot(height ~ weight, data = HowellnonAdults)
```




```
plotFun(a + b * log(x) ~ x, a = coef(linearmodel)["a"], b = coef(linearmodel  )["b"], add = TRUE, col =
```



(b)Begin with this plot: R code plot(height ~ weight , data=Howell1 col=col.alpha(rangi2,0.4)) Then use samples from the quadratic approximate posterior of the model in (a) to superimpose on the plot: (1) the predicted mean height as a function of weight, (2) the 97% HPDI for the mean, and (3) the 97% HPDI for predicted heights.

```
model <- map(
  alist(
    height ~ dnorm(mu, sigma),
    mu <- a + b1 * log(weight) ,
    a ~ dnorm(178, 100),
    b1 ~ dnorm(0, 10),
    sigma ~ dunif(0, 50)
  ),
  data = Howell1
)
```

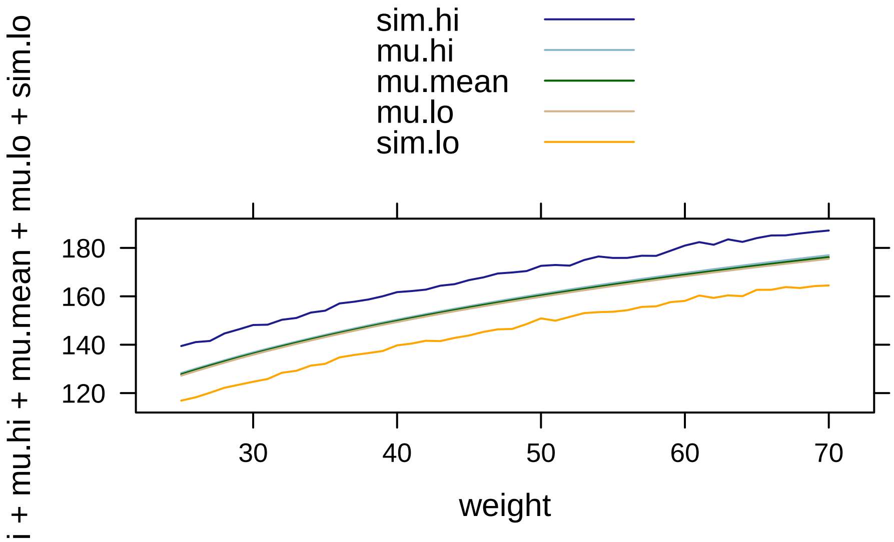
```r
model.pred <-
  data_frame(
    weight = seq(from = 25, to = 70, by = 1)

  )
mu <- link(model, data = model.pred)
```

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

```r
sim.height <- sim(model, data = model.pred)
```

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

```r
model.pred <-
  model.pred %>%
  mutate(
    mu.mean = apply(mu, 2, mean),
    mu.lo = apply(mu, 2, HPDI,prob=0.97)[1,],
    mu.hi = apply(mu, 2, HPDI,prob=0.97)[2,],
    sim.lo = apply(sim.height, 2, HPDI,prob=0.97)[1,],
    sim.hi = apply(sim.height, 2, HPDI,prob=0.97)[2,]
    )
xyplot(sim.hi + mu.hi + mu.mean + mu.lo + sim.lo ~ weight,
       data = model.pred,  type = "l", auto.key = list(lines = TRUE, points = FALSE))
```




```
plotPoints(height ~ weight, data = Howell1, col = rangi2, alpha = 0.5, add = TRUE)
```