# Statistical Rethinking 4H2

*Kara Fitze*

*May 29th, 2018*

### 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?

To begin, we load the data recorded in the !Kung census. The Howell1 data is assigned to the variable $d$, and then $d2$ is created as a subset of $d$ such that the d$age is less than 18.

```
# Load the data recorded in the !Kung census
library(rethinking)
data(Howell1)
d <- Howell1
d2 <- d[d$age < 18,]
```

We want to fit a linear regression to the data in $d2$, so the function MAP is called (to fit the model using the method of maximum a posteriori). Inside the MAP call, we provide the model definition within alist, and tell MAP which data we want it to model. The height of the population is distributed normally, with mean $\mu$ and standard deviation sigma. We define $\mu$ to be the linear model of some intercept $a$, and the weight variable (within $d2$) multiplied by some slope $b$. $a$ is distributed normally with a very weak prior (mean 0 and standard deviation 100). This prior should be overwhelmed by the sample size, so the strength of it does not matter too much here (remember that there is a huge range for children heights between 1 month and 17 years old, therefore it is difficult to select a more informative prior).$b$ is distributed normally with mean 0 and standard deviation 10. Lastly, $\sigma$ is distributed as a uniform distribution, only acquiring positive values. The data is of course $d2$.

```
model <- map(
  alist(
        height ~ dnorm(mu , sigma) ,
        mu <- a + b*weight ,
        a ~ dnorm(0 , 100) ,
        b ~ dnorm(0 , 10) ,
        sigma ~ dunif(0 , 50)
        ) ,
  data = d2)
```

To view the model, we call precis (in the rethinking package).

```
precis(model)
```

```
##         Mean StdDev  5.5% 94.5%
## a     58.22   1.40 55.99 60.45
## b      2.72   0.07  2.61  2.83
## sigma  8.44   0.43  7.75  9.13
```

The question we were asked to answer was, for every ten units of increase in weight, how much taller does the model predict a child gets? Taking the model estimate for $b$ and multiplying its value by 10 will answer this. My model assumes that as a child increases by ten kilograms, they should gain about 27 or so cm in height.

```
coef(model)['b'] * 10
```

```
##        b
## 27.20463
```

(b) Plot the raw data, with height on the vertical axis and weight on the horizontal axis. Super-impose the MAP regression line and 89% HPDI for the mean. Also superimpose the 89% HPDI for predicted heights.

To get a better sense of the weight range, lets look at the minimum and maximum weight values.Then create a sample of weights from a sequence that includes the range of true weights in $d2$.

```
min(d2$weight)
```

```
## [1] 4.252425
```

```
max(d2$weight)
```

```
## [1] 44.73551
```

```
weight <- seq(3,45,1)
```

Next we use the link function to compute the value of $\mu$ at each sample weight for each case in the data. Since we do not define $n$, we use the default of 1000 samples for each value of weight.

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

To determine the mean MAP regression line, we take each column of the matrix $\mu$ and apply the mean function. In the apply function $\mu$ is the matrix we want to use, 2 is set as the margin value, and mean is the function we want to apply.Similarly, to find an 89% HPDI for the mean, we apply the HPDI function to the columns of matrix $\mu$ and specify an 89% interval.

```
mu.mean <- apply(mu , 2 , mean)
mu.HPDI <- apply(mu, 2, HPDI, prob = 0.89)
```

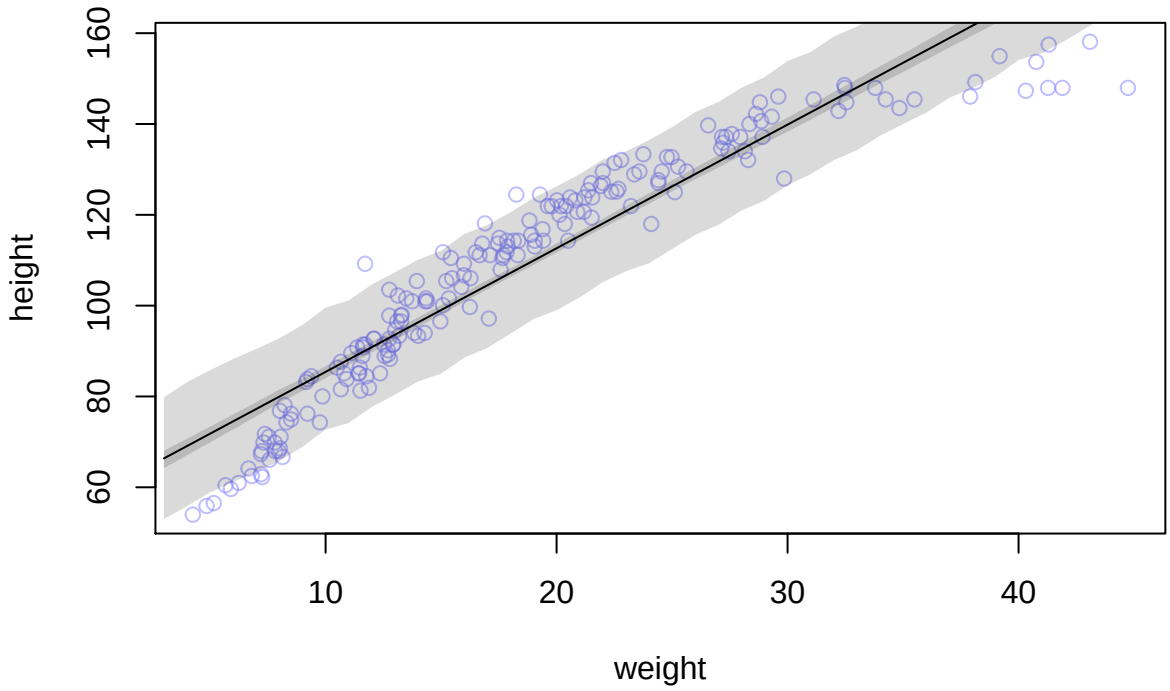
We sample from the posterior distribution and assign the samples to the variable post. Then to create simulated heights from the posterior distribution, we use the simplified apply function, applying an rnorm to each of the sample weights from the weight vector. This is done to approximate an 89% HPDI interval for simulated heights. The 89% HPDI interval is assigned to height.HPDI.

```
post <- extract.samples(model)
sim.height <- sapply( weight , function(weight){
           rnorm(
                 n = nrow(post) ,
                 mean = post$a + post$b*weight ,
                 sd=post$sigma )} )
height.HPDI <- apply(sim.height , 2 , HPDI , prob=0.89)
```

Finally, to plot the raw data with the mean MAP regression line and the 89% HPDI intervals for both the mean and simulated heights, we call plot on height and weight from $d2$. Then we use lines to plot mu.mean for each value of weight. In addition, shade is called on the objects mu.HPDI and height.HPDI with weight being the limit for both.

```
# Plot raw data
# Fading out points to make line and interval more visible
plot( height ~ weight , data=d2 , col=col.alpha(rangi2,0.5) )
# Plot the MAP line, aka the mean mu for each weight
lines(weight , mu.mean )
# Plot a shaded region for 89% HPDI
```

```
shade(mu.HPDI , weight)
shade(height.HPDI, weight)
```




(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 appears to be doing a bad job of, and what you hypothesize would be a better model.

The linear fit concerns me. This data appears to have more of a parabolic trend, and so I don't believe choosing a linear model is an accurate decision.Notice that especially in the top right corner, the raw data tends to bend while the model continues in a straight path. I would try to use a quadratic fit instead.