Statistical Rethinking Winter 2020 – Homework Week 2

Alessandro Gentilini .@gmail (just auditing)

December 7, 2020

1. The weights listed below were recorded in the !Kung census, but heights were not recorded for these individuals. Provide predicted heights and 89% compatibility intervals for each of these individuals. That is, fill in the table below, using model-based predictions.

Individual	weight	expected height	89% interval
1	45		
2	40		
3	65		
4	31		

I create the model:

```
## R code 4.42
# load data again, since it's a long way back
library(rethinking)
```

```
## Loading required package: rstan
```

```
## Loading required package: StanHeaders
```

```
## Loading required package: ggplot2
```

```
## rstan (Version 2.21.2, GitRev: 2e1f913d3ca3)
```

```
## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
```

```
## Loading required package: parallel
```

```
## rethinking (Version 2.13)
```

```
##
## Attaching package: 'rethinking'
```

```
## The following object is masked from 'package:stats':
##
##     rstudent
```

```
data(Howell1); d <- Howell1; d2 <- d[ d$age >= 18 , ]

# define the average weight, x-bar
xbar <- mean(d2$weight)
```

```
# fit model
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 )
```

And then I simulated the heights for the four individuals:

```
weights <- c(45,40,65,31)
sim.height <- sim(m4.3,data=list(weight=weights))
```

Since the *expected* height is requested I will compute the mean (also known as *expectation* or *expected value*):

```
expected.height <- apply(sim.height,2,mean)
```

and then I compute the PI:

```
PIs <- apply(sim.height,2,PI,prob=.89)
```

I then create a dataframe just for the sake of printing a table.

```
answer <- data.frame(
    individual=1:4,
    weight=weights,
    expected.height=expected.height,
    low=PIs[1,],
    high=PIs[2,])
```

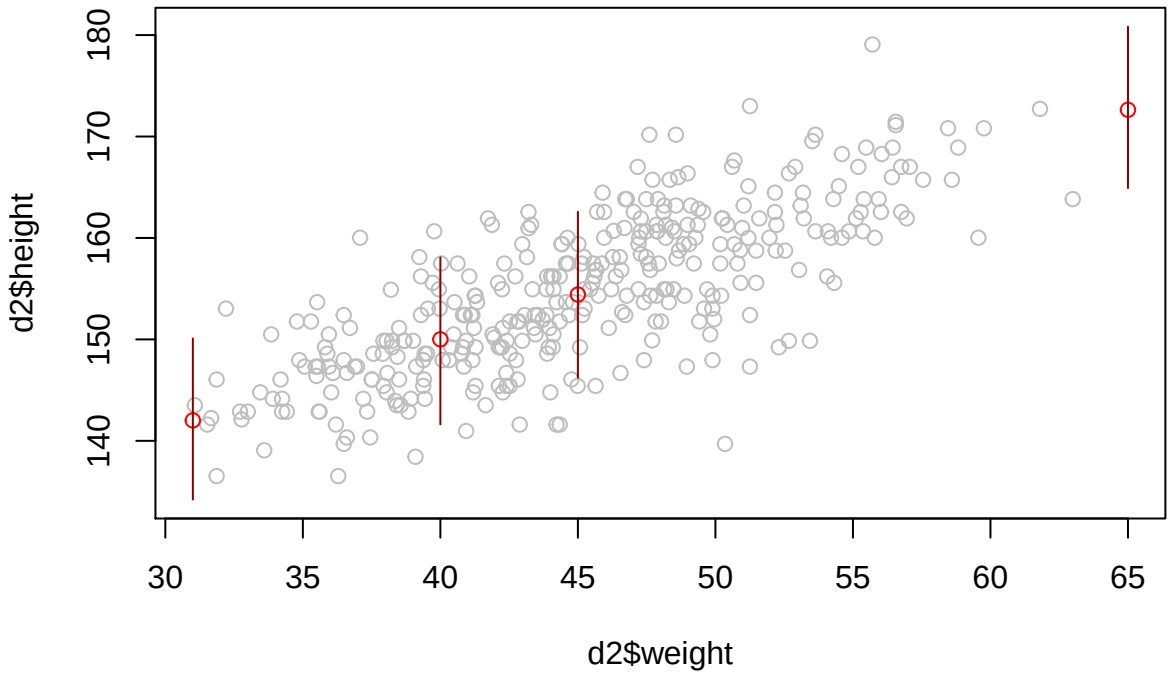
I then print the table rounding the results to the centimeter since it is meaningless to report the millimeter when the uncertainties are in the range of the decimeters.

```
knitr::kable(answer,digits=0,
    col.names=c('Individual','weight','expected height',
                'lower 89% PI','upper 89% PI'))
```

Individual	weight	expected height	lower 89% PI	upper 89% PI
1	45	154	146	163
2	40	150	142	158
3	65	173	165	181
4	31	142	134	150

And then I plot the simulated heights above all the recorded data:

```
x.min=min(c(d2$weight,answer$weight))
x.max=max(c(d2$weight,answer$weight))
y.min=min(c(d2$height,answer$height,answer$low))
y.max=max(c(d2$height,answer$height,answer$high))
plot(d2$weight,d2$height,type='p',col='grey',xlim=c(x.min,x.max),ylim=c(y.min,y.max))
points(answer$weight,answer$expected.height,type='p',col='red')
for(i in answer$individual){
    lines(x=c(answer$weight[i],answer$weight[i]),y=c(answer$low[i],answer$high[i]),col='darkred')
}
```


d2$height
140
150
160
170
180
30
35
40
45
50
55
60
65
d2$weight

2. Model the relationship between height (cm) and the natural logarithm of weight (log-kg): `log(weight)`. Use the entire `Howell1` data frame, all 544 rows, adults and non-adults. Use any model type from Chapter 4 that you think useful: an ordinary linear regression, a polynomial or a spline. I recommend a plain linear regression, though. Plot the posterior predictions against the raw data.

I compute the new model:

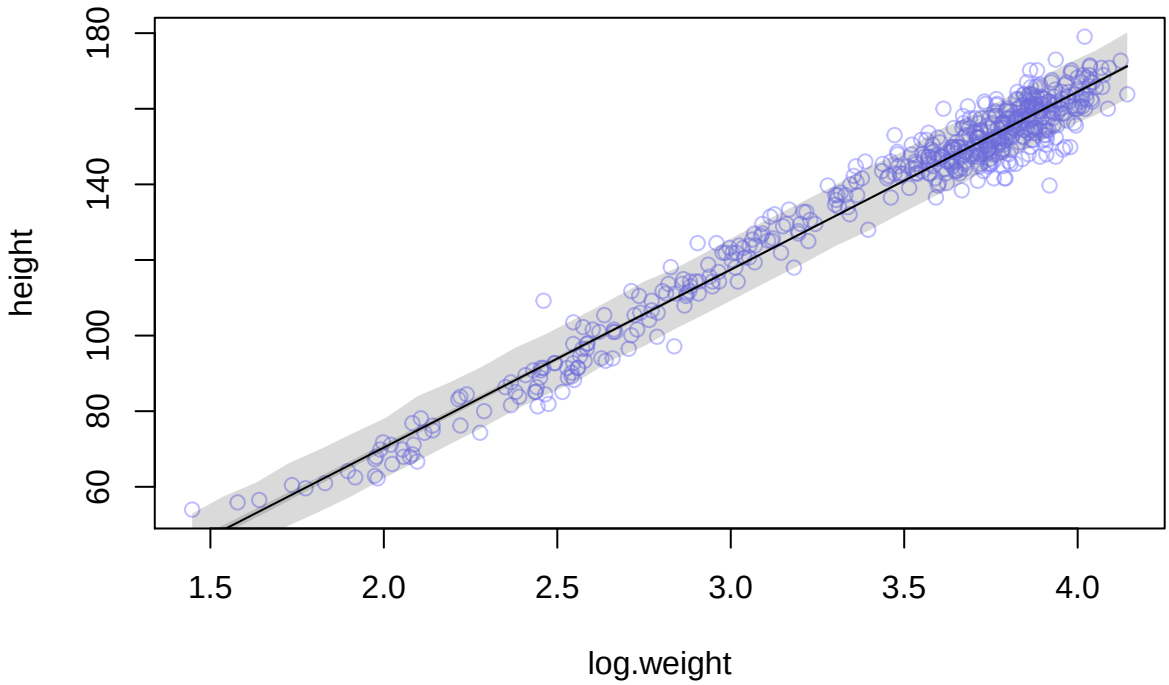
```
d$log.weight <- log(d$weight)

# define the average log weight, x-bar
log.xbar <- mean(d$log.weight)

# fit model
week_2_hw <- quap(
    alist(
        height ~ dnorm( mu , sigma ) ,
        mu <- a + b*( log.weight - log.xbar ) ,
        a ~ dnorm( 178 , 20 ) ,
        b ~ dlnorm( 0 , 1 ) ,
        sigma ~ dunif( 0 , 50 )
    ) , data=d )
```

I simulate some heights:

```
## R code 4.67
log.weight.seq <- seq( from=min(d$log.weight) , to=max(d$log.weight) , length.out=30 )
pred_dat <- list( log.weight=log.weight.seq )
mu <- link( week_2_hw , data=pred_dat )
mu.mean <- apply( mu , 2 , mean )
mu.PI <- apply( mu , 2 , PI , prob=0.89 )
sim.height <- sim( week_2_hw , data=pred_dat )
height.PI <- apply( sim.height , 2 , PI , prob=0.89 )
```

I plot the the posterior predictions against the raw data.

```
## R code 4.68
plot( height ~ log.weight , d , col=col.alpha(rangi2,0.5) )
lines( log.weight.seq , mu.mean )
shade( mu.PI , log.weight.seq )
shade( height.PI , log.weight.seq )
```


height
60
80
100
140
180
1.5
2.0
2.5
3.0
3.5
4.0
log.weight

3. Plot the prior predictive distribution for the polynomial regression model in Chapter 4. You can modify the the code that plots the linear regression prior predictive distribution. 20 or 30 parabolas from the prior should suffice to show where the prior probability resides. Can you modify the prior distributions of α, β_1, and β_2 so that the prior predictions stay within the biologically reasonable outcome space? That is to say: Do not try to fit the data by hand. But do try to keep the curves consistent with what you know about height and weight, before seeing these exact data.

After some tinkering (including in my opinion an insuccess when I forced β_2 to be positive using for it a `rlnorm`) I found this solution that I like:

```
## R code 4.38
set.seed(2971)
N <- 30                    # 30 parabolas
a <- rnorm( N , 178 , 20 )
b1 <- rlnorm( N , 0 , 1 )
b2 <- rnorm( N , 0 , 1/4 )



## R code 4.39
plot( NULL , xlim=range(d2$weight) , ylim=c(-100,400) ,
    xlab="weight" , ylab="height" )
abline( h=0 , lty=2 )
abline( h=272 , lty=1 , lwd=0.5 )
mtext( "b1 ~ rlnorm( N , 0 , 1 ), b2 ~ rnorm( N , 0 , 1/4 )" )
xbar <- mean(d2$weight)
for ( i in 1:N ) curve( a[i] + b1[i]*(x - xbar) + b2[i]*(x-xbar)*(x-xbar),
    from=min(d2$weight) , to=max(d2$weight) , add=TRUE ,
    col=col.alpha("black",0.2) )
```