

assignemt_2_blaufuss

Dennis Blaufuss

11/10/2021

1. Consider the following model from the textbook to use the !Kung census to predict height from weight of adults.

```
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
)
```

Using this model, provide the predicted heights and 89% credibility intervals for each of the following individuals:

```
weight.seq <- c(46, 61, 35, 52, 56)
sim.height <- sim(m4.3, data = list(weight = weight.seq), n=1e4)
height.hat <- apply(sim.height, 2, mean)
height.PI <- apply(sim.height, 2, PI, prob = 0.89)
```

#	individual	weight	expected height	4.5%	94.5%
#	1	46	155.4683	147.1811	163.4612
#	2	61	169.0215	160.7459	177.2698
#	3	35	145.5769	137.3297	153.7493
#	4	52	160.9799	152.8080	169.1208
#	5	56	164.4176	156.2995	172.5035

2. Plot the prior predictive distribution for the polynomial regression model in Chapter 4. Use `extract.prior` to inspect the prior, and modify the code that simulates and plots prior predictive distributions for linear regression to perform prior predictive simulations. Plotting between 30 and 50 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 outcomes? You should not attempt to fit the data by hand. Instead, try to keep the curves consistent with what you know about height and weight before seeing the !Kung data.

```

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, 100),
    sigma ~ dunif(0, 50)
  ),
  data = d
)

weight.seq <- seq(from=-2.5, to=2.5, length.out=30)
pred_dat <- list(weight_s=weight.seq, weight_s2=weight.seq^2)

prior <- extract.prior(m4.5)

```

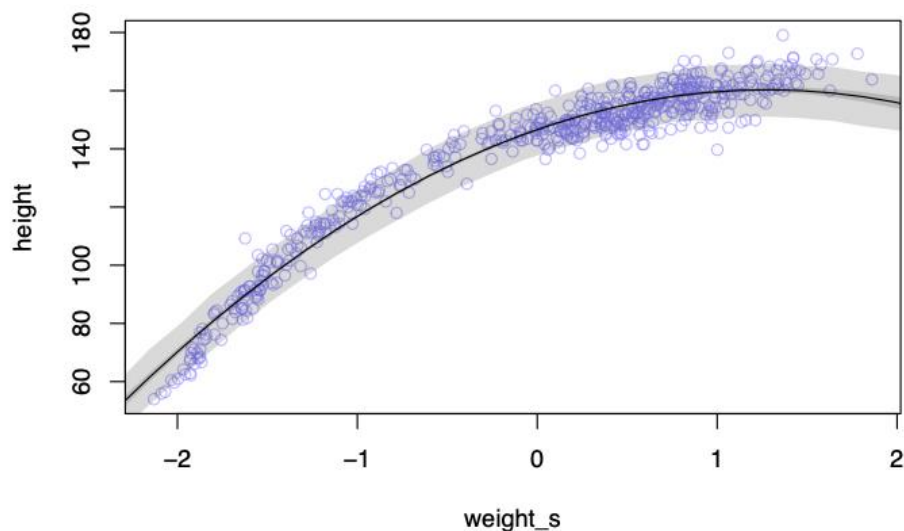
Plotting:

```

mu <- link(m4.5, data=pred_dat)
mu.mean <- apply(mu, 2, mean)
mu.PI <- apply(mu, 2, PI, prob=0.89)
sim.height <- sim(m4.5, data=pred_dat)
height.PI <- apply(sim.height, 2, PI, prob=0.89)

plot(height ~ weight_s, d, col=col.alpha(rangi2, 0.5))
lines(weight.seq, mu.mean)
shade(mu.PI, weight.seq)
shade(height.PI, weight.seq)

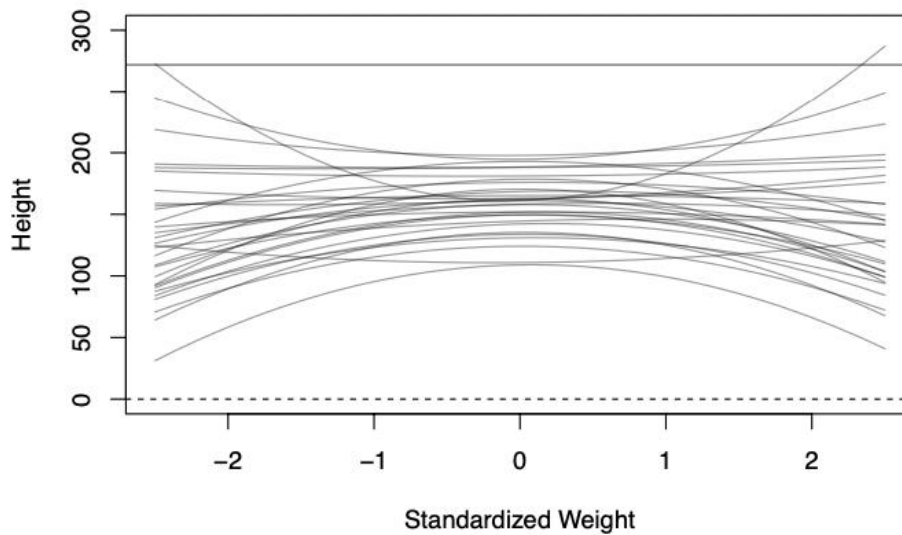
```



Modification for prior distribution:

```
mu <- link(m4.5, post=prior, data=pred_dat)
plot(NULL, xlim=range(weight.seq), ylim=c(0, 300), xlab="Standardized Weight",
, ylab="Height")

for(i in 1:30) {
  lines(weight.seq, mu[i,], col=col.alpha("black", 0.4))
}
abline(h=0, lty=2)
abline(h=272, lty=1, lwd=0.5)
```



3. Write down a multiple regression to evaluate the claim: 1 The price of houses in Frankfurt is linearly related to size, but only after controlling for location (i.e., postal code). You only need to write down the model definition. There are 41 postal codes in Frankfurt. For this exercise, consider houses to belong to one of four postal code regions: 603, 604, 605, and 659.

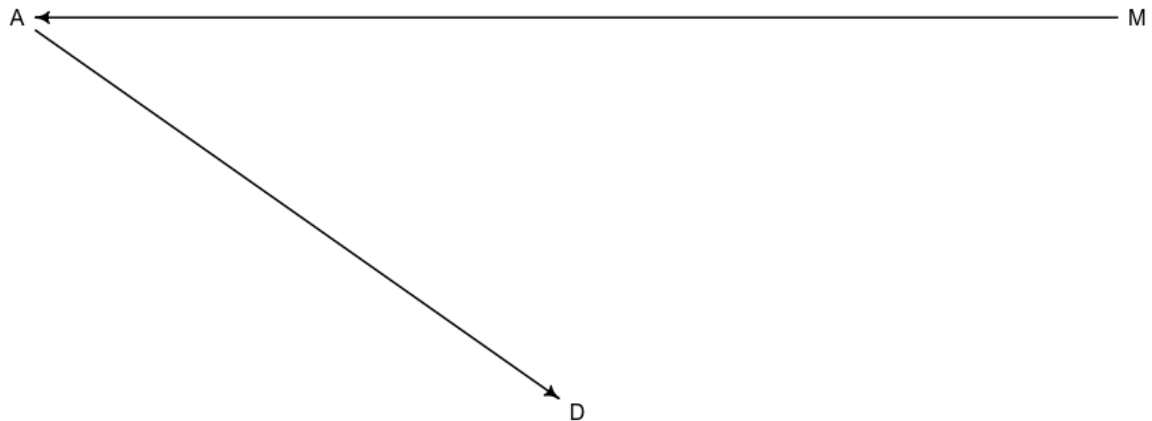
$$P_i \sim \text{Normal}(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta_s * S_i + \beta_{603} * 603_i + \beta_{604} * 604_i + \beta_{605} * 605_i + \beta_{659} * 659_i$$

with S = Size, P = Price and β = weights for each Zip code.

4. In the divorce example, suppose the DAG is: $M \rightarrow A \rightarrow D$.

```
dag <- dagitty("dag{ M -> A -> D}")
coordinates(dag) <- list( x=c(A=0,D=1,M=2) , y=c(A=0,D=1,M=0))
drawdag(dag)
```



What are the implied conditional independencies of this graph?

```
impliedConditionalIndependencies(dag)
# D _||_ M | A
```

D and M are conditionally independent under the condition of knowing A.

Are the data consistent with it?

```
equivalentDAGs(dag)

#[[1]]
#dag {
#A [pos="0.000,0.000"]
#D [pos="1.000,1.000"]
#M [pos="2.000,0.000"]
#A -> D
#M -> A
#}

#[[2]]
#dag {
#A [pos="0.000,0.000"]
#D [pos="1.000,1.000"]
#M [pos="2.000,0.000"]
#A -> D
#A -> M
#}

#[[3]]
#dag {
```

```
#A [pos="0.000,0.000"]  
#D [pos="1.000,1.000"]  
#M [pos="2.000,0.000"]  
#A -> M  
#D -> A  
#}
```

Thus data is consistent.