Statistical Rethinking: Homework Problems

Melissa Van Bussel

June 12, 2018

5H3

This time we want to consider 2 new models: one will be weight ~ avgfood + groupsize, and the other will be weight ~ avgfood + groupsize + area. We then want to compare these two models to the ones we had before, as well as decide whether avgfood or area is the better predictor if we had to choose only one, supporting our decision with any plots/tables that we may need.

Finally, after fitting the model, we'll observe that when both avgfood and area are included in the model, their standard errors are larger than when they are included in separate models, and their effects are essentially reduced (close to 0). We can go ahead and explain this one right now since it's easy: They're collinear.

```
library(rethinking)
data(foxes)
d <- foxes
cor(d) # as expected, high correlation
```

```
##                group     avgfood  groupsize       area      weight
## group      1.0000000  0.34084493  0.4009594 0.37777286 -0.15190033
## avgfood    0.3408449  1.00000000  0.9014829 0.88310378 -0.02503892
## groupsize  0.4009594  0.90148290  1.0000000 0.82759448 -0.16099376
## area       0.3777729  0.88310378  0.8275945 1.00000000  0.01947728
## weight    -0.1519003 -0.02503892 -0.1609938 0.01947728  1.00000000
```

Now let's answer the first part. First, we create the two new models. I'll be using very conservative priors, since we don't really know how much each predictor should contribute to the overall model.

```
# Model containing only avgfood and groupsize
lmod5h3_1 <- map(
  alist(
    weight ~ dnorm(mu, sigma),
    mu <- a + ba * avgfood + bg * groupsize,
    a ~ dnorm(0, 10),
    ba ~ dnorm(0, 10),
    bg ~ dnorm(0, 10),
    sigma ~ dunif(0, 10)
  ), data = d
)

# Model containing avgfood, groupsize, and area
lmod5h3_2 <- map(
  alist(
    weight ~ dnorm(mu, sigma),
    mu <- a + bt * area + bg * groupsize + ba * avgfood,
    a ~ dnorm(0, 10),
    bt ~ dnorm(0, 10),
    bg ~ dnorm(0, 10),
    ba ~ dnorm(0, 10),
    sigma ~ dunif(0, 10)
```

```
  ), data = d
)
```

Now, we can take a look at the models we've just created.

```
precis(lmod5h3_1, prob = 0.95)
```

```
##        Mean StdDev  2.5% 97.5%
## a      4.13   0.43  3.29  4.97
## ba     3.79   1.20  1.43  6.14
## bg    -0.56   0.16 -0.87 -0.26
## sigma  1.12   0.07  0.97  1.26
```

```
precis(lmod5h3_2, prob = 0.95)
```

```
##        Mean StdDev  2.5% 97.5%
## a      4.06   0.43  3.23  4.90
## bt     0.39   0.24 -0.08  0.86
## bg    -0.60   0.16 -0.91 -0.30
## ba     2.47   1.44 -0.35  5.29
## sigma  1.10   0.07  0.96  1.25
```

In the first model, all of the predictors look "important" (or whatever word you might want to use), but in the second model, **avgfood** and **area** don't look "important", since the intervals include 0. We expected this to happen - the question told us this ahead of time, plus we've already observed that these two predictors are collinear.

To gain a deeper understanding, we can also compare our results to the models calculated by classical OLS, and see if these predictors are significant in the classical sense.

```
summary(lm(weight ~ avgfood + groupsize, data = d))
```

```
##
## Call:
## lm(formula = weight ~ avgfood + groupsize, data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.98506 -0.67290 -0.06745  0.73525  2.96652
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.1232     0.4380   9.414 6.94e-16 ***
## avgfood       3.8275     1.2291   3.114 0.002338 **
## groupsize    -0.5687     0.1584  -3.589 0.000492 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.131 on 113 degrees of freedom
## Multiple R-squared:  0.1029, Adjusted R-squared:  0.08703
## F-statistic: 6.481 on 2 and 113 DF,  p-value: 0.002164
```

```
summary(lm(weight ~ avgfood + groupsize + area, data = d))
```

```
##
## Call:
## lm(formula = weight ~ avgfood + groupsize + area, data = d)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -2.61759 -0.70325 -0.08013  0.59766  3.11292
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.0638     0.4367   9.305 1.33e-15 ***
## avgfood       2.5089     1.4787   1.697 0.092530 .
## groupsize    -0.6077     0.1593  -3.815 0.000224 ***
## area          0.3850     0.2436   1.581 0.116722
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.124 on 112 degrees of freedom
## Multiple R-squared:  0.1225, Adjusted R-squared:  0.09899
## F-statistic: 5.211 on 3 and 112 DF,  p-value: 0.002093
```

As expected, when both **avgfood** and **area** are included in the same model, neither are significant. This happened because these two predictors are collinear. Since they're collinear, we now want to decide which one of the two would be better to include in the model. There are various ways od doing this, but one such way is by plotting each one of the predictors we're interested in while keeping the others constant at their means (similar to in question 5H2).

```
# Set up: Plot weight ~ avgfood while keeping groupsize and area constant at their means
avgfood_sequence <- seq(0, round(max(d$avgfood) + 1))
avgfood_prediction <- data.frame(avgfood = avgfood_sequence,
                                 groupsize = mean(d$groupsize),
                                 area = mean(d$area))
mu_avgfood <- link(lmod5h3_2, avgfood_prediction)
```

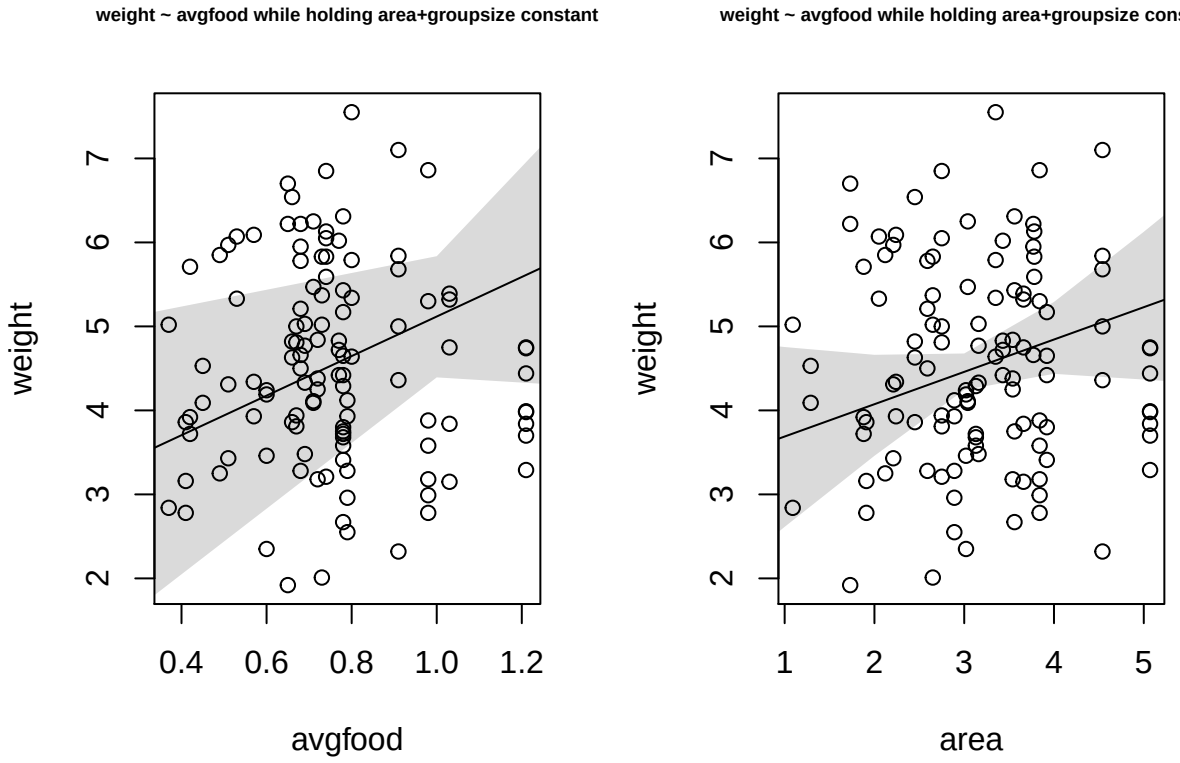
```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```
mu_avgfood_mean <- apply(mu_avgfood, 2, mean)
mu_avgfood_PI <- apply(mu_avgfood, 2, PI, prob = 0.95) # use 0.95 since 5H1 did

# Set up: Plot weight ~ area while keeping groupsize and avgfood constant at their means
area_sequence <- seq(0, round(max(d$area) + 1))
area_prediction <- data.frame(area = area_sequence,
                              groupsize = mean(d$groupsize),
                              avgfood = mean(d$avgfood))
mu_area <- link(lmod5h3_2, area_prediction)
```

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

```
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```
mu_area_mean <- apply(mu_area, 2, mean)
mu_area_PI <- apply(mu_area, 2, PI, prob = 0.95) # use 0.95 since 5H1 did

# Actually plot them now
par(mfrow = c(1, 2))
plot(weight ~ avgfood, data = d, main = "weight ~ avgfood while holding area+groupsize constant",
     cex.main = 0.6)
lines(avgfood_sequence, mu_avgfood_mean)
shade(object = mu_avgfood_PI, lim = avgfood_sequence)
plot(weight ~ area, data = d, main = "weight ~ avgfood while holding area+groupsize constant",
     cex.main = 0.6)
lines(area_sequence, mu_area_mean)
shade(object = mu_area_PI, lim = area_sequence)
```



Visually, it appears like weight ~ area is better, since the interval is tighter. We can also take a look at classical OLS regression to see if the two methods agree.

```
summary(lm(weight ~ avgfood + groupsize, data = d))
```

```
##
## Call:
## lm(formula = weight ~ avgfood + groupsize, data = d)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -2.98506 -0.67290 -0.06745  0.73525  2.96652
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.1232     0.4380   9.414 6.94e-16 ***
## avgfood       3.8275     1.2291   3.114 0.002338 **
## groupsize    -0.5687     0.1584  -3.589 0.000492 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.131 on 113 degrees of freedom
## Multiple R-squared:  0.1029, Adjusted R-squared:  0.08703
## F-statistic: 6.481 on 2 and 113 DF,  p-value: 0.002164
```

```
summary(lm(weight ~ area + groupsize, data = d))
```

```
##
## Call:
## lm(formula = weight ~ area + groupsize, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3479 -0.7307 -0.1385  0.6808  3.0643
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.4502     0.3758  11.843  < 2e-16 ***
## area          0.6182     0.2028   3.048 0.002866 **
## groupsize    -0.4326     0.1224  -3.535 0.000591 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.133 on 113 degrees of freedom
## Multiple R-squared:  0.09994,    Adjusted R-squared:  0.08401
## F-statistic: 6.273 on 2 and 113 DF,  p-value: 0.002609
```

We can see that both of these models are pretty terrible, but the first one is slightly better, which disagrees with the conclusion we made earlier. This tells us that there are perhaps better ways to measure goodness of fit when it comes to Bayesian models, and in fact, that's what chapter 6 is all about.

One last note: If you recall from the correlation matrix, **avgfood** and **groupsize** were even more highly correlated with each other than **avgfood** and **area** were, which explains why neither one of these models were very good. We should probably only be including one of the 3 predictors in the model if we want to see some improvement.