Chapter 5 Questions

Melissa Van Bussel

June 20, 2018

Chapter 5 points: 2 + 8 + 6 = 16 points total.

Easy Questions (2 points total)

5E1

(1) is single linear regression, so not (1). (2) is missing an intercept term, meaning the intercept term is 0. This is known as regression through the origin, and is indeed multiple linear regression. (3) and (4) are also multiple linear regression.

5E2

$$d_i \sim \text{Normal}(\mu, \sigma)$$
$$\mu_i = \alpha + \beta_l l_i + \beta_p p_i$$

5E3

Both slope parameters should be on the right side of zero since the question says they are both positively correlated. The model can be represented by

$$d_i \sim \text{Normal}(\mu, \sigma)$$
$$\mu_i = \alpha + \beta_f f_i + \beta_p p_i$$

5E4

(1) is equivalent (the intercept term ends up being for C), and (3) is equivalent for the same reason but the intercept term is now for A instead of C.

Medium Questions (8 points total)

5M1 (2 points)

An example of a spurious correlation would be: (suppose) Per capita consumption of caviar correlates with % of population that voted Conservative

Upon first glance, one might think that a Conservative government gives people free caviar or something, when in reality, rich people can afford caviar and rich people also tend to vote conservative. The spurious correlation here would be average annual income.

5M2 (2 points)

A masked relationship is where two predictor variables are both correlated with the outcome variable, but one is positively correlated and the other is negatively correlated, so they appear to "cancel each other out". In addition, the two predictor variables are positively correlated with each other.

For example… the outcome variable could be BMI. One predictor would be weight, and the other predictor would be height. Since BMI = weight/height, having a higher weight will positively correlate with BMI while having a higher height will negatively correlate with BMI. Obviously, height and weight are positively correlated with each other.

5M3 (2 points)

A high divorce rate might cause a higher marriage rate because people who get divorced become single and are eligible to marry again, increasing the overall marriage rate.

5M5 (2 points)

Outcome: Obesity Rate Mechanism 1: Less driving = more exercise Mechanism 2: Less driving = less eating out

Another predictor that would lead to more exercise and less eating out would be: - number of sports played - number of days per month that the person goes to the gym - basically anything that leads people to live healthy, active lives.

Hard Questions (6 points total)

The 3 hard exercises use the foxes data from the rethinking package. This dataset contains information on the vulpes vulpes species of fox. The 5 varaibles are:

(1) group: Number of social group that individual fox belongs to
(2) avgfood: Average amount of food available in the territory
(3) groupsize: # of foxes in the social group
(4) area: Size of the territory
(5) weight: Body weight of the individual fox

```
library(rethinking)
```

```
## Loading required package: rstan
## Warning: package 'rstan' was built under R version 3.3.3
## Loading required package: ggplot2
## Warning: package 'ggplot2' was built under R version 3.3.3
## Loading required package: StanHeaders
## Warning: package 'StanHeaders' was built under R version 3.3.3
## rstan (Version 2.17.3, 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 1.59)
```

```
data(foxes)
d <- foxes
head(d)
```

```
##   group avgfood groupsize area weight
## 1     1    0.37         2 1.09   5.02
## 2     1    0.37         2 1.09   2.84
## 3     2    0.53         2 2.05   5.33
## 4     2    0.53         2 2.05   6.07
## 5     3    0.49         2 2.12   5.85
## 6     3    0.49         2 2.12   3.25
```

5H1 (2 points)

We want to fit 2 regressions using map. The first is body weight as a function of territory size, and the second is body weight as a function of group size. Then we want to plot the results with the MAP regression line and 95% interval of the mean. Finally, we determine whether either of the variables are important (on their own) for predicting body weight.

First, let's look at the distribution of the two predictors we're interested in, so we can determine what values to use in our regressions:

```
# let's just use the rounded values
round(mean(d$area))
```

```
## [1] 3
```

```
round(sd(d$area))
```

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

```
round(mean(d$groupsize))
```

```
## [1] 4
```

```
round(sd(d$groupsize))
```

```
## [1] 2
```

```
# Compute the two models
lmod1 <- map(
  alist(
    weight ~ dnorm(mu, sigma),
    mu <- a + bt * area,
    a ~ dnorm(3, 3),
    bt ~ dnorm(0, 5),
    sigma ~ dunif(0, 10)
  ), data = d
)

lmod2 <- map(
  alist(
    weight ~ dnorm(mu, sigma),
    mu <- a + bg * groupsize,
    a ~ dnorm(4, 4),
    bg ~ dnorm(0, 5),
    sigma ~ dunif(0, 10)
  ), data = d
)
```

```r
# Compute the interval to shade on the plot
area_sequence <- seq(from = 0, to = round(max(d$area)) + 1, length.out = 100)
mu1 <- link(lmod1, data = data.frame(area = area_sequence))
```

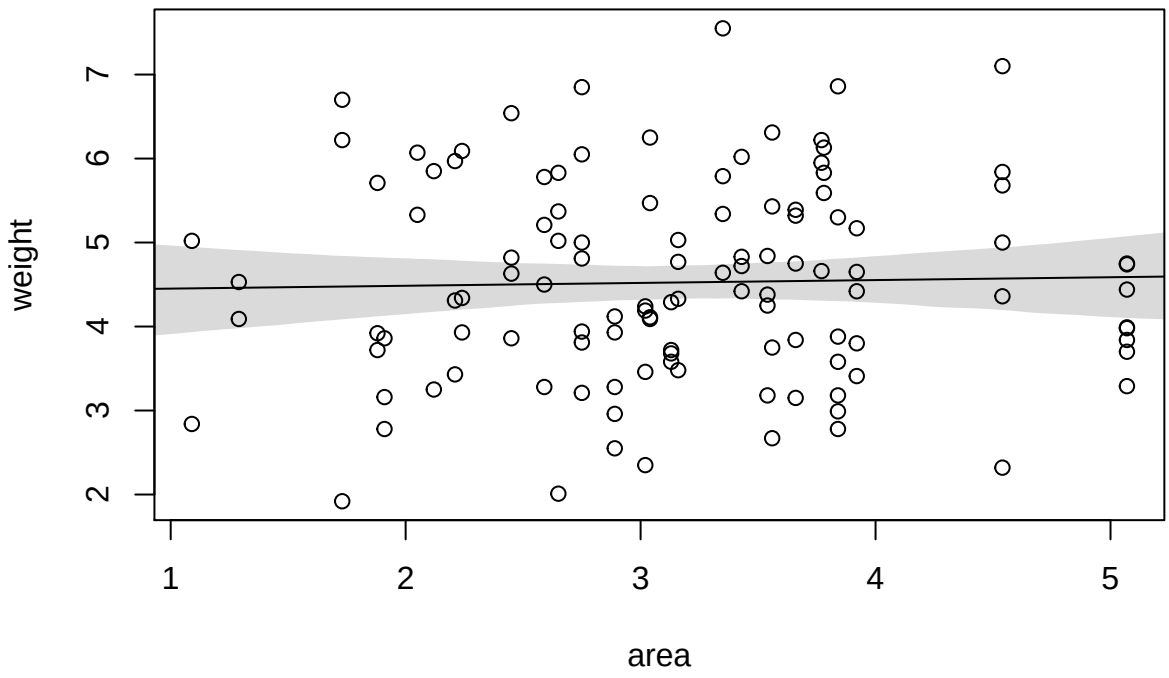
```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```r
groupsize_sequence <- seq(from = 0, to = round(max(d$groupsize)) + 1, length.out = 100)
mu2 <- link(lmod2, data = data.frame(groupsize = groupsize_sequence))
```

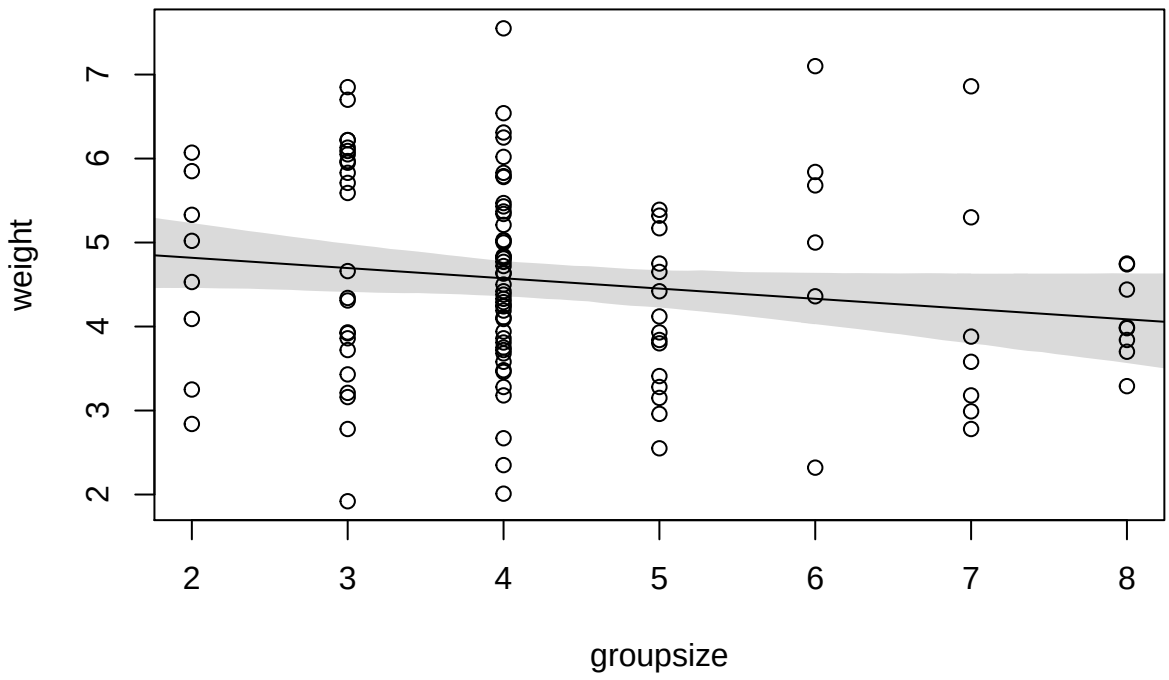
```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```r
mu1_mean <- apply(mu1, 2, mean)
mu2_mean <- apply(mu2, 2, mean)
mu1_95PI <- apply(mu1, 2, PI, prob = 0.95)
mu2_95PI <- apply(mu2, 2, PI, prob = 0.95)

# Plot results
plot(weight ~ area, data = d)
lines(area_sequence, mu1_mean)
shade(mu1_95PI, area_sequence)
```




```
plot(weight ~ groupsize, data = d)
lines(groupsize_sequence, mu2_mean)
shade(mu2_95PI, groupsize_sequence)
```




```
# Determine whether either variable is important to the model on its own
precis(lmod1)
```

```
##       Mean StdDev  5.5% 94.5%
## a     4.43   0.39  3.81  5.05
## bt    0.03   0.12 -0.16  0.22
## sigma 1.18   0.08  1.06  1.30
```

```
precis(lmod2)
```

```
##        Mean StdDev  5.5% 94.5%
## a      5.06   0.32  4.54  5.58
## bg    -0.12   0.07 -0.23 -0.01
## sigma  1.16   0.08  1.04  1.29
```

```
# Neither seems significant just from looking at the values
# But we can also compute classical OLS to see if either predictor is significant
summary(lm(weight ~ area, data = d)) # just as expected, not significant
```

```
##
## Call:
## lm(formula = weight ~ area, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5739 -0.7942 -0.1326  0.8494  3.0158
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.45093    0.39426  11.289   <2e-16 ***
## area         0.02484    0.11943   0.208    0.836
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.189 on 114 degrees of freedom
## Multiple R-squared:  0.0003794,  Adjusted R-squared:  -0.008389
## F-statistic: 0.04326 on 1 and 114 DF,  p-value: 0.8356
```

```
summary(lm(weight ~ groupsize, data = d)) # again just as expected, not significant
```

```
##
## Call:
## lm(formula = weight ~ groupsize, data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.77628 -0.79568 -0.07958  0.87707  2.97762
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.06798    0.32773  15.464   <2e-16 ***
## groupsize   -0.12390    0.07114  -1.742   0.0843 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.174 on 114 degrees of freedom
## Multiple R-squared:  0.02592,    Adjusted R-squared:  0.01737
## F-statistic: 3.033 on 1 and 114 DF,  p-value: 0.08426
```

5H2 (2 points)

This time we want to fit a multiple linear regression with weight as the response and area and groupsize as the predictors. Then we want to plot the predictions of the model for each predictor, while holding the other predictor constant, and compare to the results we got in the previous question. We should then answer why we get different results using this method as opposed to the method in the previous question.

```
# Compute the model
lmod5h2 <- map(
  alist(
    weight ~ dnorm(mu, sigma),
    mu <- a + bt * area + bg * groupsize,
    a ~ dnorm(0, 100), # We have no idea what the mean/sd are this time, so make conservative estimate
    bt ~ dnorm(0, 10),
    bg ~ dnorm(0, 10),
    sigma ~ dunif(0, 10)
  ), data = d
)

# Take a look at the model
precis(lmod5h2) # This time, both predictors look significant
```

```
##        Mean StdDev  5.5% 94.5%
## a      4.45   0.37  3.86  5.04
```

```
## bt     0.62   0.20  0.30  0.94
## bg    -0.43   0.12 -0.63 -0.24
## sigma  1.12   0.07  1.00  1.24
```

```
# Compare to classical OLS to see if they're significant in that model too
summary(lm(weight ~ area + groupsize, data = d)) # Both significant, as expected
```

```
##
## 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
```

```
# Plot weight ~ area while holding groupsize constant at its mean
area_sequence <- seq(0, round(max(d$area) + 1))
area_prediction <- data.frame(area = area_sequence,
                              groupsize = mean(d$groupsize))
mu_area <- link(lmod5h2, area_prediction)
```

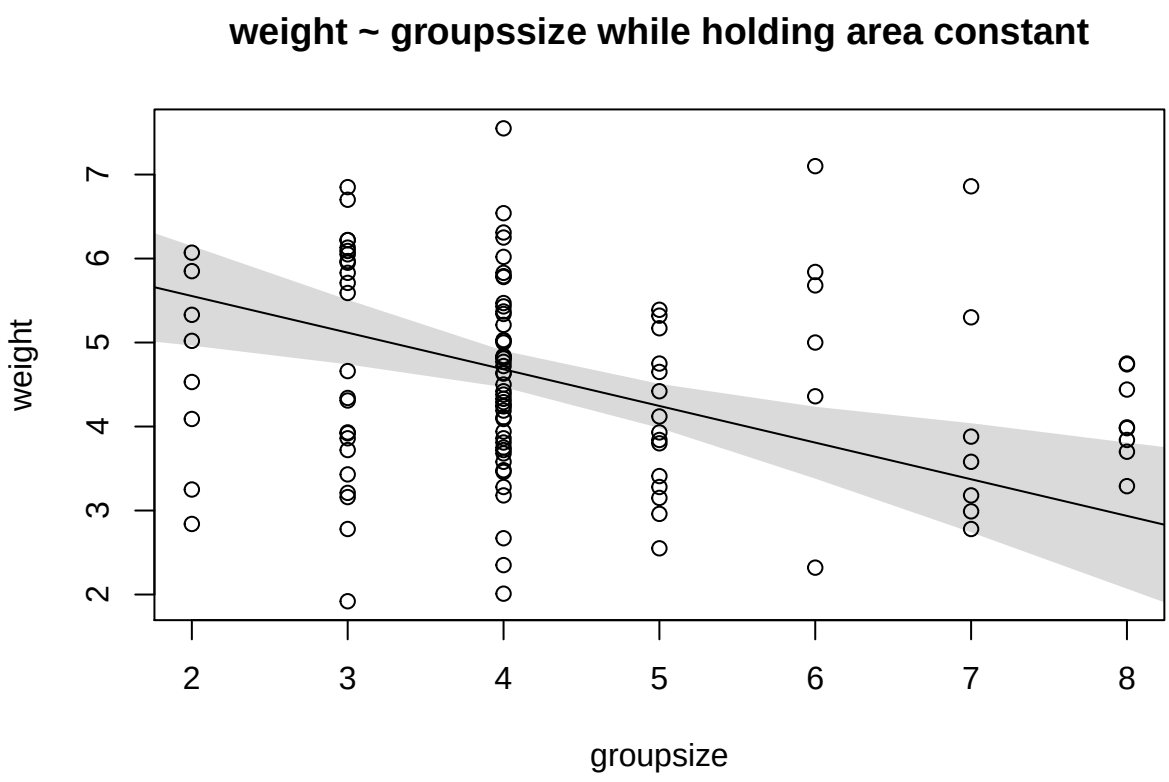
```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```
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
plot(weight ~ area, data = d, main = "weight ~ area while holding groupsize constant")
lines(area_sequence, mu_area_mean)
shade(object = mu_area_PI, lim = area_sequence)
```


```
# Plot weight ~ groupsize while holding area constant at its mean
groupsize_sequence <- seq(0, round(max(d$groupsize) + 1))
groupsize_prediction <- data.frame(groupsize = groupsize_sequence,
                                   area = mean(d$area))
mu_groupsize <- link(1mod5h2, groupsize_prediction)
```

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

```
mu_groupsize_mean <- apply(mu_groupsize, 2, mean)
mu_groupsize_PI <- apply(mu_groupsize, 2, PI, prob = 0.95) # use 0.95 since 5H1 did
plot(weight ~ groupsize, data = d, main = "weight ~ groupssize while holding area constant")
lines(groupsize_sequence, mu_groupsize_mean)
shade(object = mu_groupsize_PI, lim = groupsize_sequence)
```




```
# Alone, neither one was significant, but together, they were both significant.
```

5H3 (2 points)

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)[4, 2] # as expected, high correlation
```

```
## [1] 0.8831038
```

Now let's answer the first part:

```
# Model containing only avgfood and groupsize
lmod5h3_1 <- map(
  alist(
    weight ~ dnorm(mu, sigma),
    mu <- a + ba * avgfood + bg * groupsize,
    a ~ dnorm(0, 10), # We have no idea what the mean/sd are this time, so make conservative estimate
```

```
    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, 100), # We have no idea what the mean/sd are this time, so make conservative estimate
    bt ~ dnorm(0, 10),
    bg ~ dnorm(0, 10),
    ba ~ dnorm(0, 10),
    sigma ~ dunif(0, 10)
  ), data = d
)

# Take a look at the models
precis(lmod5h3_1)
```

```
##        Mean StdDev  5.5% 94.5%
## a      4.13   0.43  3.44  4.82
## ba     3.79   1.20  1.86  5.71
## bg    -0.56   0.16 -0.81 -0.31
## sigma  1.12   0.07  1.00  1.23
```

```
precis(lmod5h3_2)
```

```
##        Mean StdDev  5.5% 94.5%
## a      4.07   0.43  3.39  4.76
## bt     0.39   0.24  0.01  0.77
## bg    -0.60   0.16 -0.85 -0.35
## ba     2.46   1.44  0.16  4.75
## sigma  1.10   0.07  0.99  1.22
```

```
# They all look significant, in the first one, but avgfood and area don't look significant in the 2nd
# let's see what classical OLS thinks about this
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
```

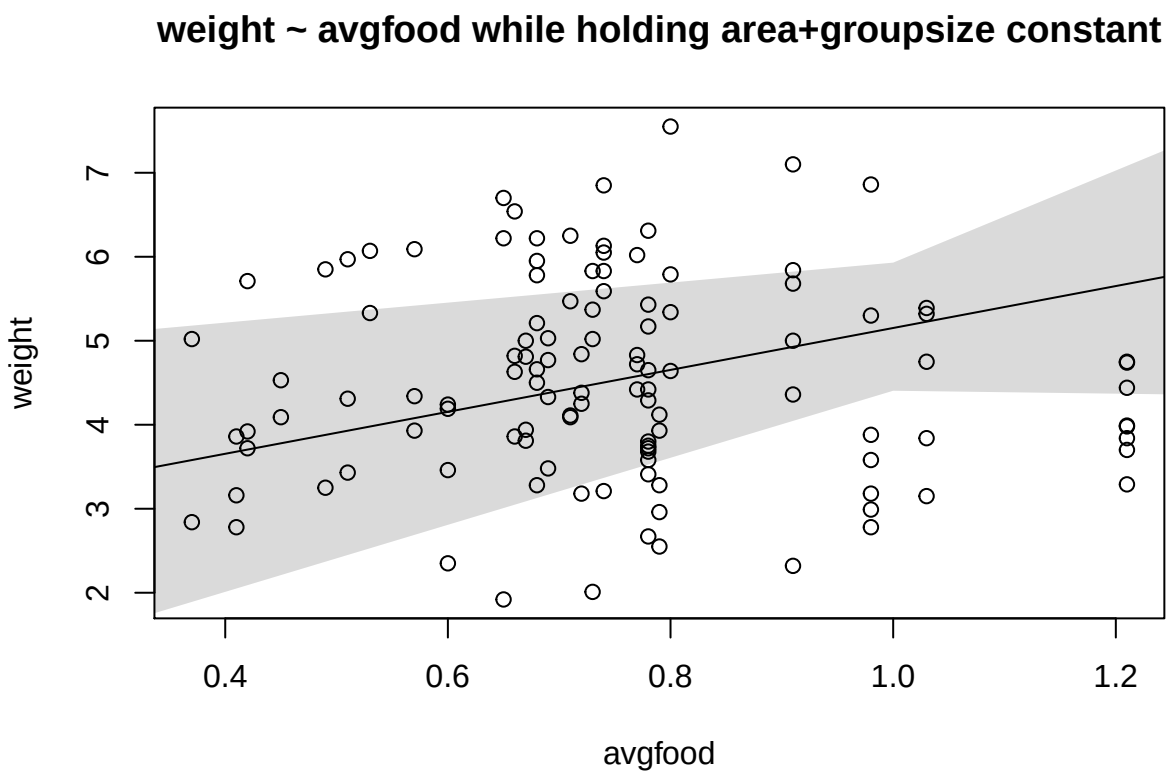
```
# As expected, when both avgfood and area are included in the same model, neither are significant
# This happened since they are collinear

# Since they're collinear, we want to decide which one of the two would be better to include
# in the model
# We do this by plotting each one of the predictors we're interested in while keeping the others
# constant at their means

# 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
plot(weight ~ avgfood, data = d, main = "weight ~ avgfood while holding area+groupsize constant")
lines(avgfood_sequence, mu_avgfood_mean)
shade(object = mu_avgfood_PI, lim = avgfood_sequence)
```

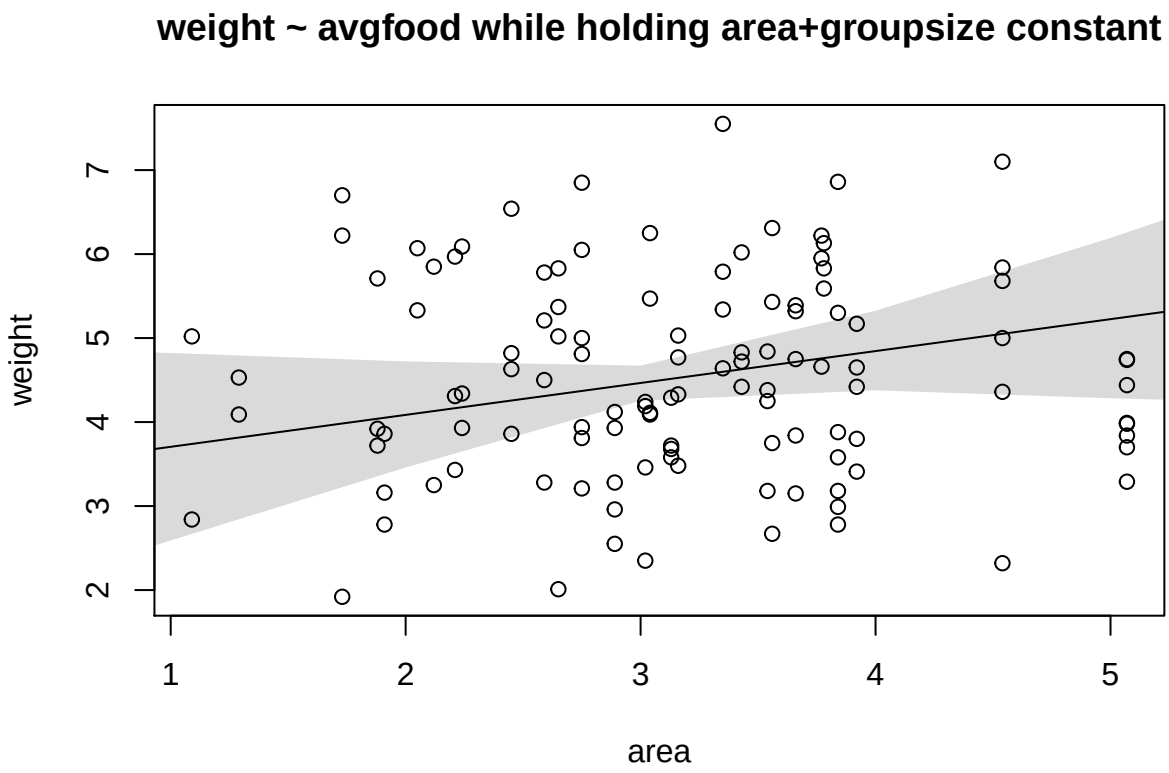



```
# Plot weight ~ area while keeping groupsize and avgfood constant at their means
area_sequence <- seq(0, round(max(d$area) + 1))
area_prediction <- data.frame(area = area_sequence,
                              groupsize = mean(d$groupsize),
                              avgfood = mean(d$avgfood))
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
plot(weight ~ area, data = d, main = "weight ~ avgfood while holding area+groupsize constant")
```

```
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
# Let's confirm using OLS, even though we haven't learned about goodness of fit in this class yet
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
```

```r
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
```

```r
# Both are fairly crappy, but we can see that using area is just slightly better.
```