EEEB UN3005/GR5005
Lab - Week 07 - 11 and 13 March 2019

Xun Zhao, xz2827

Multiple Regression Models

For this week's lab, we'll be using data on foxes occupying urban England. You can access this data once you've loaded the `rethinking` package with the code `data(foxes)`. You can get a brief summary of all the variables contained in the dataset using `?foxes` or a data frame summary function of your choice.

Note, for all models you'll be asked to construct here, assume a prior of `dnorm(0, 10)` for all intercept and slope parameters and a prior of `dunif(0, 10)` for all standard deviation parameters. Furthermore, start values of 0 for all intercept and slope parameters and 5 for all standard deviation parameters should ensure good model fits throughout these exercises.

Exercise 1: Linear Regression with Territory Size as a Predictor

Construct a linear regression model of fox body weight (`weight` variable) using territory size (`area` variable) as a predictor.

After fitting the model, use `precis()` to display the 90% PIs for all model parameters. Plot the results of this regression model, displaying the MAP regression line and the 95% interval of the mean.

What does this model suggest about the effect of territory size on fox body weight?

```
data(foxes)
d = foxes
model1 = map(
    alist(
        weight ~ dnorm(mu, sigma),
        mu <- a + b * area,
        a ~ dnorm(0, 10),
        b ~ dnorm(0, 10),
        sigma ~ dunif(0, 10)),
    start = list(a = 0, b = 0, sigma = 5),
    data = d)
precis(model1, prob = 0.9)
```

```
##       Mean StdDev  5.0% 95.0%
## a     4.44   0.39  3.80  5.09
```

```
## b     0.03   0.12 -0.17  0.22
## sigma 1.18   0.08  1.05  1.31
```

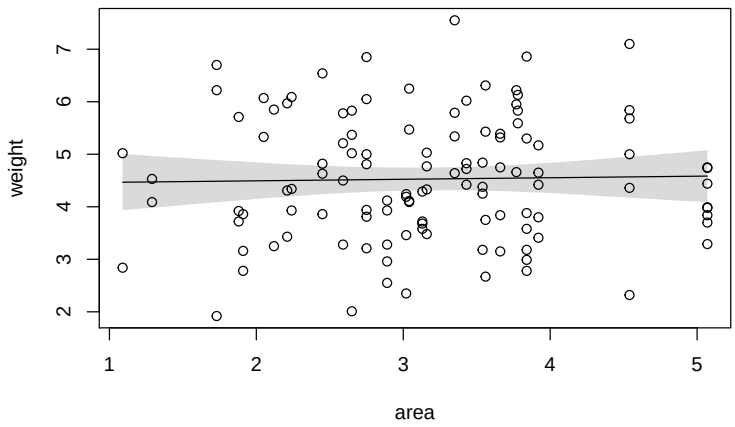
```
area.seq = seq(from = min(d$area), to = max(d$area), length.out = 100)
pred.data = data.frame(area = area.seq)

mu = link(model1, data = pred.data, n = 10000)
```

```
## [ 1000 / 10000 ]
[ 2000 / 10000 ]
[ 3000 / 10000 ]
[ 4000 / 10000 ]
[ 5000 / 10000 ]
[ 6000 / 10000 ]
[ 7000 / 10000 ]
[ 8000 / 10000 ]
[ 9000 / 10000 ]
[ 10000 / 10000 ]
```

```
mu.mean = apply(mu, 2, mean)
mu.PI = apply(mu, 2, PI, prob = 0.95)

plot(weight ~ area, data = d)
lines(area.seq, mu.mean)
shade(mu.PI, area.seq)
```



Answer:

The `weight` variable and `area` variable are not linear correlated with each other. The regression line is almost flat.

Exercise 2: Linear Regression with Group Size as a Predictor

Now, construct a linear regression model of fox body weight using fox group size (`groupsize` variable) as a predictor.

After fitting the model, use `precis()` to display the 90% PIs for all model parameters. Plot the results of this regression model, displaying the MAP regression line and the 95% interval of the mean.

What does this model suggest about the effect of group size on fox body weight?

```
model2 = map(
    alist(
        weight ~ dnorm(mu, sigma),
        mu <- a + b * groupsize,
        a ~ dnorm(0, 10),
        b ~ dnorm(0, 10),
        sigma ~ dunif(0, 5)),
    #start = list(a = 0, b = 0, sigma = 5),
    data = d)
precis(model2, prob = 0.9)
```

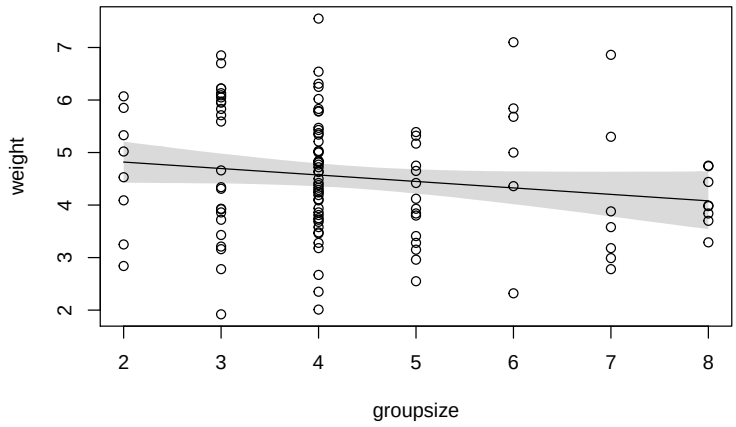
```
##        Mean StdDev  5.0% 95.0%
## a      5.06   0.32  4.53  5.60
## b     -0.12   0.07 -0.24 -0.01
## sigma  1.16   0.08  1.04  1.29
```

```
group.seq = seq(min(d$groupsize), max(d$groupsize), length.out = 100)
pred.data = data.frame(groupsize = group.seq)
mu = link(model2, data = pred.data, n = 10000)
```

```
## [ 1000 / 10000 ]
[ 2000 / 10000 ]
[ 3000 / 10000 ]
[ 4000 / 10000 ]
[ 5000 / 10000 ]
[ 6000 / 10000 ]
[ 7000 / 10000 ]
[ 8000 / 10000 ]
[ 9000 / 10000 ]
[ 10000 / 10000 ]
```

```
mu.mean = apply(mu, 2, mean)
mu.PI = apply(mu, 2, PI, prob = 0.95)
plot(weight ~ groupsize, data = d)
```

```
lines(group.seq, mu.mean)
shade(mu.PI, group.seq)
```



Answer:

`weight` is slightly negatively correlated with `groupsize`.

Exercise 3: Multiple Regression with Both Predictors

Now fit a multiple linear regression with fox body weight as the outcome and both territory size and group size as predictor variables.

What does this model say about the importance of each predictor variable? Why do you get different results than you got in the exercises just above?

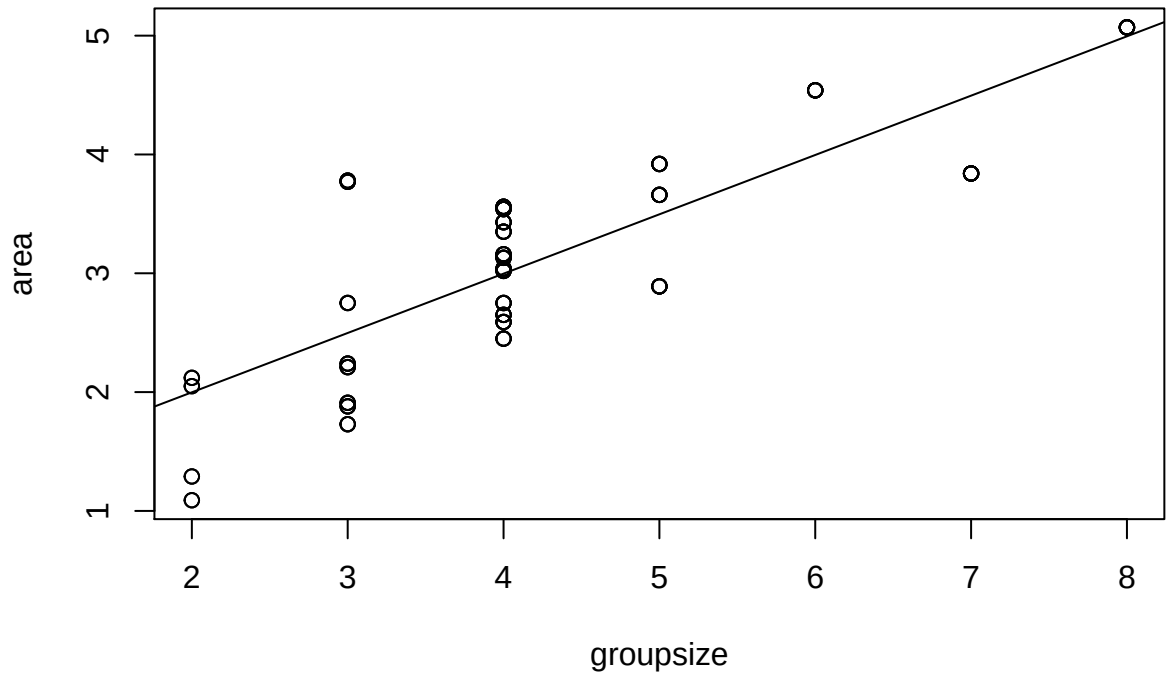
```
model3 = map(
    alist(
        weight ~ dnorm(mu, sigma),
        mu <- a + b.area * area + b.gsize * groupsize,
        a ~ dnorm(0, 10),
        b.area ~ dnorm(0, 10),
        b.gsize ~ dnorm(0, 10),
        sigma ~ dunif(0, 5)),
    #start = list(a = 0, b.area = 0, b.gsize = 0, sigma = 5),
    data = d)
precis(model3, prob = 0.9)
```

```
##          Mean StdDev  5.0% 95.0%
```

```
## a         4.44   0.37  3.83  5.05
## b.area    0.62   0.20  0.29  0.95
## b.gsize -0.43   0.12 -0.63 -0.23
## sigma     1.12   0.07  1.00  1.24
```

```r
mean.area = mean(d$area)
mean.gsize = mean(d$groupsize)

plot(area ~ groupsize, data = d)
abline(lm(area ~ groupsize, data = d))
```



Answer:

The multiple regression shows that `weight` is positively correlated with `area` and negatively correlated with `groupsize`. And the slope of `area` becomes larger and the slope of `groupsize` becomes smaller to more negative.

The difference between multiple regression and single variable regression is because of the ***masking***. As the plot shows, `area` is positively correlated with `groupsize`, so their effects to `weight` is counteracted with each other.

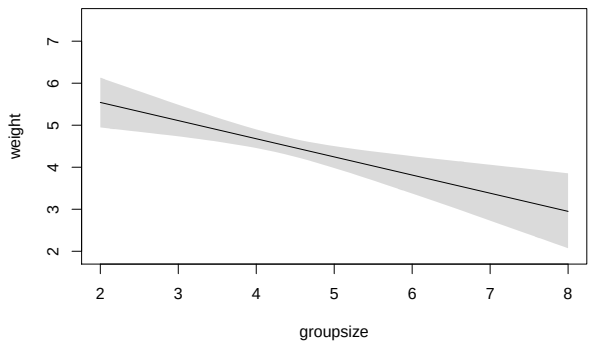
Exercise 4: Counterfactual Plots for a Multiple Regression

For the multiple regression model, plot predictions by showing the 95% interval of the mean for each predictor, holding the other predictor constant at its mean value.

```
pred.data = data.frame(area = mean.area, groupsize = group.seq)
mu = link(model3, data = pred.data, n = 10000)
```

```
## [ 1000 / 10000 ]
[ 2000 / 10000 ]
[ 3000 / 10000 ]
[ 4000 / 10000 ]
[ 5000 / 10000 ]
[ 6000 / 10000 ]
[ 7000 / 10000 ]
[ 8000 / 10000 ]
[ 9000 / 10000 ]
[ 10000 / 10000 ]
```

```
mu.mean = apply(mu, 2, mean)
mu.PI = apply(mu, 2, PI, prob = 0.95)
plot(weight ~ groupsize, data = d, type = "n")
lines(group.seq, mu.mean)
shade(mu.PI, group.seq)
```




```
pred.data = data.frame(area = area.seq, groupsize = mean.gsize)
mu = link(model3, data = pred.data, n = 10000)
```

```
## [ 1000 / 10000 ]
[ 2000 / 10000 ]
[ 3000 / 10000 ]
[ 4000 / 10000 ]
[ 5000 / 10000 ]
[ 6000 / 10000 ]
```

```
[ 7000 / 10000 ]
[ 8000 / 10000 ]
[ 9000 / 10000 ]
[ 10000 / 10000 ]
```

```
mu.mean = apply(mu, 2, mean)
mu.PI = apply(mu, 2, PI, prob = 0.95)
plot(weight ~ area, data = d, type = "n")
lines(area.seq, mu.mean)
shade(mu.PI, area.seq)
```