Lab 4

Stephen R. Proulx, Taom Sakal

Bayesian Statistical Modeling Winter 2024

Lab Exercise, Week 4

When is this lab due? Labs are due on the Thursday after they are assigned. However, in many cases you can complete them during the lab period itself. This assignment is due on Thursday, 2/8/2024.

Multiple Regression

We'll use a dataset included with the Rethinking package, called "foxes". This dataframe includes observations of the weight of 116 foxes from 30 groups in urban areas of England. These foxes are territorial, and individuals are able to forage within their groups territory. The dataframe includes the area controlled by the group, as well as a measure of food available, labeled "avgfood" for average food. Fox groups vary in size, and this is encoded by the column "groupsize".

We'll start by using rethinking's `standardize` function on the potential predictor variables which will ensure that they have a mean of 0. This gives us the predictor variables relative to the mean, which we put in their own columns.

```
data("foxes")
d <- as_tibble(foxes)

# Put standardized values of the three potential predictors in their own column.
d$F <- standardize(d$avgfood)
d$A <- standardize(d$area)
d$G <- standardize(d$groupsize)
```

You can view the tibble before and after the standardization to make sure it worked. You already know View() can do this, but you can also *Ctrl-click* the dataframe variable in the code or press *F2* when the text cursor is on the variable. (Doing these on a function will also take you to the source code of the function. This is a nice way to see exactly where a function comes from and what it does.)

```
d
```

Let's have a look at the data. We're plotting fox weight on the y-axis and territory area on the x-axis. Since area has been standardized, a value $A = 0$ indicates the mean area in the dataset, so foxes on the left are in smaller than average territories, and foxes on the right are in larger than average territories. We'll also color the points by groupsize (non-standardized) and can see that larger groups also tend to be on the right.

```
ggplot(data = d, aes(x = A, y = weight, group = groupsize, color = groupsize)) +
  geom_point()
```

Group size (G), area controlled by the group (A), and the food available in a group's area (F) are all candidates for variables that can predict weight. To start let's fit separate linear models for each and see how they do.

Does area affect fox weight?

Let's first fit a model to see how area (A) predicts a fox's weight. In the Latex code below, write out the equations for a linear model for the effect of area on weight and choose parameters for the prior distributions. (We've partially filled this out for you. Add in the ? parts.)

$$D_i \sim \text{Normal}(?, ?)? = a + bA * A_i a \sim \text{Normal}(?, ?) bA \sim ? \sigma \sim \text{Exponential}(1)$$

Now take this model you've just written down and fit it with `quap`.

```
m1 <- quap()

precis(m1)  # Look at the summary
```

Next let's run a prior predictive simulation to check that our priors are reasonably consistent with the data. (rethinking's extract.prior function will be useful here.)

```
prior <- extract.prior(m1)
seq_A <- seq(from = -2, to = 2, length.out = 30)
samples_prior_m1 <- link_df(m1, data = list(A = seq_A), post = prior)

ggplot(data = samples_m1, aes(x = A, y = mu)) +
  geom_point(alpha = .05)
```

Now with the fitted model we can ask how well it represents the actual observations. We also ask how area values that we did not observe are represented by the model.

To do this let's first make samples of possible mu's for a given area. That is, get a list of area (a) values and for each of them pull a couple samples from the posterior. This will give you a and bA values. Then with the power of *link_df* we can calculate the corresponding *mu*. This effectively gives us samples of *mu* from the posterior.

```
seq_A <- seq(from = -2, to = 2, length.out = 30) # A list of A values
samples_post_m1 <- link_df(m1, data = list(A = seq_A))
```

Because for each A we pull multiple samples we also get multiple *mu* values. The following code makes a table that, for each area size, gives us the mean *mu* associated with it along with the bounds one the 80 percent interval.

```
summarize_samples_post_m1 <- group_by(samples_post_m1, A) %>%
  summarize(
    mean_mu = mean(mu),
    lower_mu = quantile(mu, 0.10),
    upper_mu = quantile(mu, 0.90)
  ) %>%
  ungroup()
```

With *geom_ribbon* we can plot the range that this 80% interval covers. Overlaid on this is the actual data. Remember that this 80% interval is for the *mean weight*, not the weight itself. The blue points are the mean-mu values themselves.

```
ggplot(data = d, aes(x = A, y = weight)) +
  geom_point() +
  geom_ribbon(
    data = summarize_samples_post_m1, inherit.aes = FALSE,
    aes(x = A, ymin = lower_mu, ymax = upper_mu), alpha = 0.5, fill = "blue"
  ) +
  geom_point(
    data = summarize_samples_post_m1, inherit.aes = FALSE,
    aes(x = A, y = mean_mu, ), color = "blue"
  )
```

We would also like to put actual weight predictions onto this plot, not just the average mu values. The function *sim_df* can give us these.

```
simulations_post_m1 <- sim_df(m1, data = list(A = seq_A))
```

Now remake the above plot, this time also add a red ribbon for the 80% percentile interval for the simulated weights. This ribbon should be much wider than the blue ribbon since we have additional variation caused by sampling from the average.

```
# Put plot code here
```

From these plots how well do you think that this linear model predicts fox weight?

< Type answer here >

Does food level affect fox weight?

Repeat everything you did above but now with food level (F) instead of area.

```
m2 <- quap()

precis(m2)
```

Does group size affect weight?

Repeat the analysis again, but now with group size (G) as the predictor.

```
m3 <- quap()

precis(m3)
```

Does food affect weight when we control for group size?

We've seen that food and area do not predict fox weight, and that group size somewhat predicts it. Let's try adding both at once. Fill out the model below so that *mu* comes from a linear combination of food (F) and group size (G).

$$D_i \sim ?\mu_i = ?a \sim ?bF \sim ?bG \sim ?\sigma \sim ?$$

Now use quap to fit the model.

```
m4 <- quap()


precis(m4)
```

We can plot each of the lower level parameters against each other with the *pairs* function from rethinking.

```
rethinking::pairs(m4)
```

Now make a plot that, for each group, compares the mean predicted weights (mu) actual mean weights. That is to say, each fox group has data about food and weight. We can put those values in our model to predict the mu of each group. We can then compare our prediction with the actual mean weight of each group.

```
samples_post_m4 <- # put code to get samples here

# Make a mean_weight by mean_mu plot here.
```

Finally, have the model simulate the weights themselves and make a plot of your choosing to compare them.