Statistical Rethinking 11H6

Melissa Van Bussel

July 22, 2018

11H6 (2 points)

Begin by loading in the Fish data

```
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)
```

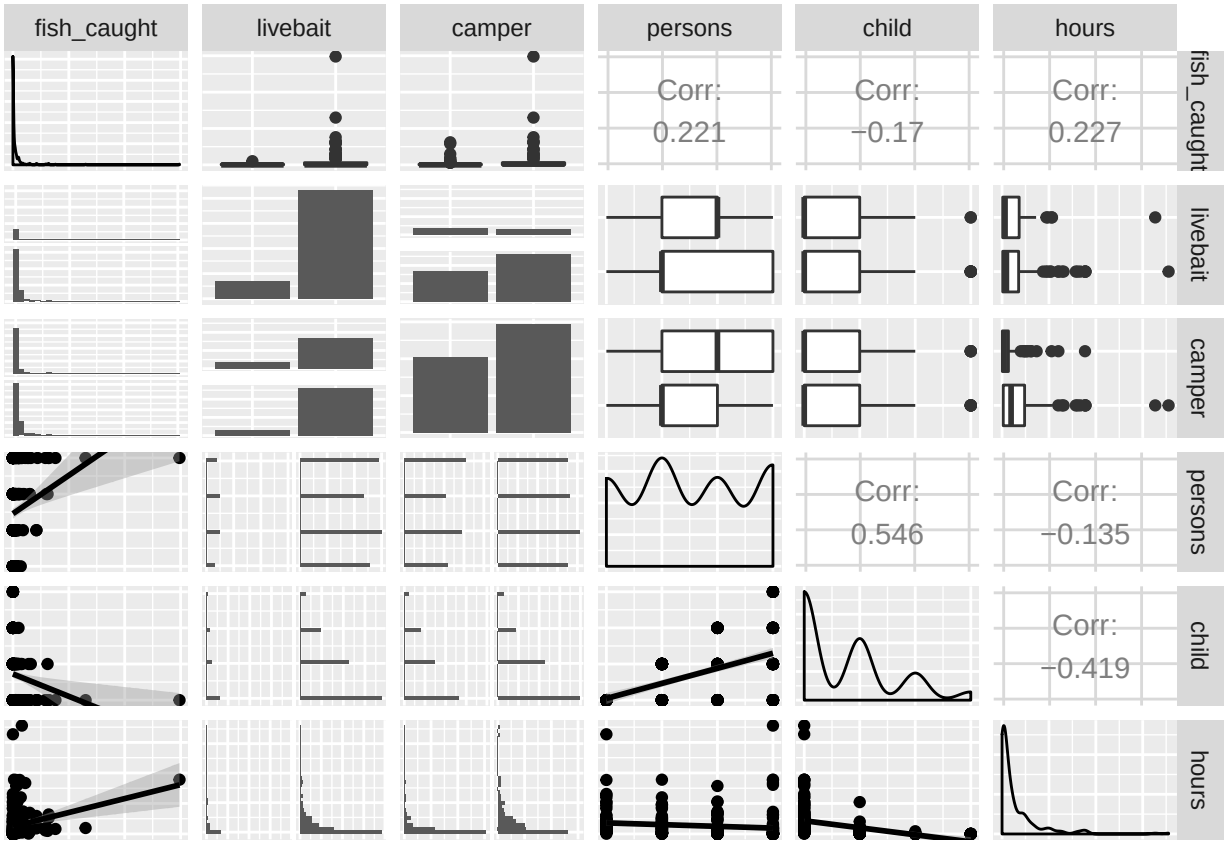
```
data(Fish)
f <- Fish
str(f)
```

The variables in this dataset are as follows:

- **fish_caught:** Number of fish caught during visit
- **livebait:** Whether or not group used livebait to fish
- **camper:** Whether or not group had a camper
- **persons:** Number of adults in group
- **child:** Number of children in group
- **hours:** Number of hours group spent in park

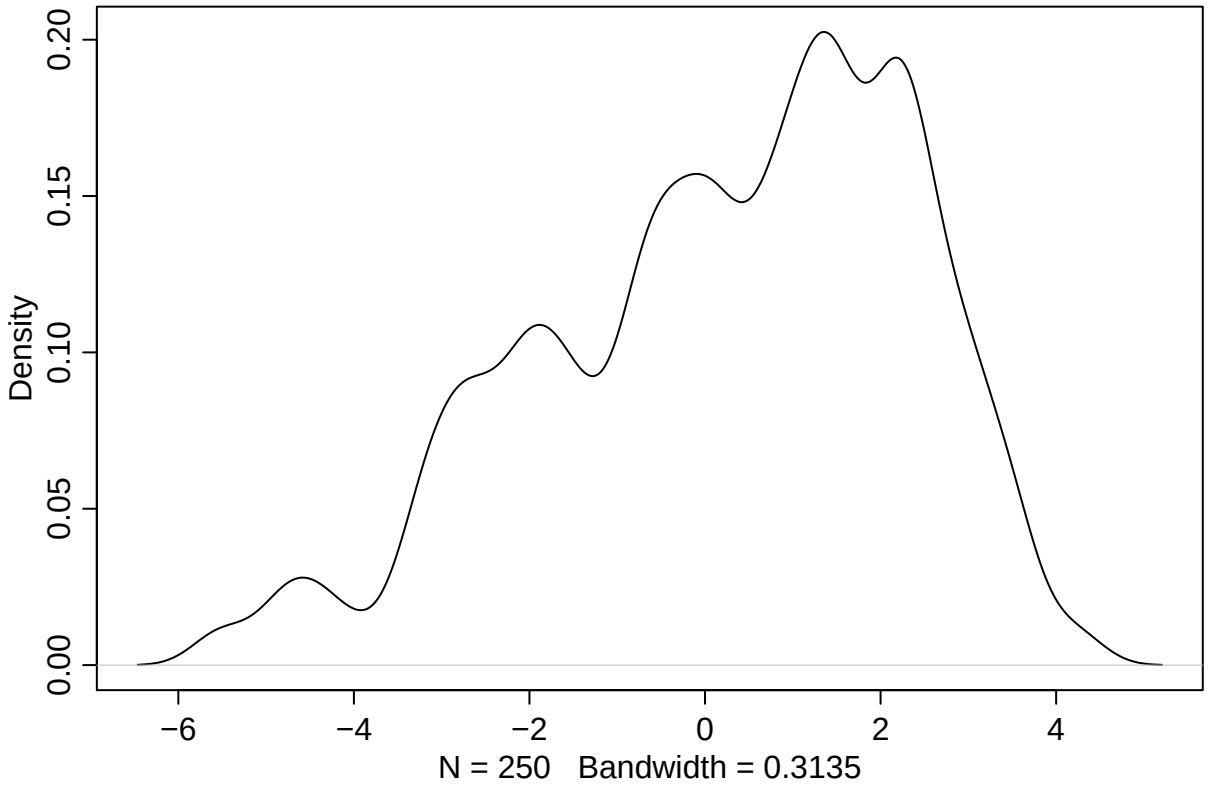
The **fish_caught** variable will be our response, and the other variables are candidates for predictors. Let's take a look at the distributions of the other variables so we can make a better decision about which predictors to include in our model. (Note that in order to do this, we'll need to create a second copy of the data frame where we convert the factor variables to factors so that **ggpairs()** can interpret properly.)

```
library(ggplot2)
library(GGally)
f_factor <- f
f_factor$livebait <- as.factor(f$livebait)
f_factor$camper <- as.factor(f$camper)
ggpairs(f_factor, lower = list(continuous = "smooth"), diag = list(continuous = "density"),
    axisLabels = "none")
```



We can see that the distribution of the **hours** variable looks like it could use a logarithmic transformation in order to be more normal.

```
f$hours <- log(f$hours)
dens(f$hours)
```



This still doesn't look very Normal, but it looks better than it did before we applied the transformation.

Since the question doesn't ask us to use specific predictors, we have some freedom to experiment and see which predictors create the best model. Personally, I think that **camper** is the only variable there which probably doesn't matter much at all. The rest seem like they could have some sort of effect on how many fish are caught by a group of campers. We can compute a couple of models and compare them by using model diagnostics. Either way, since this is a zero-inflated dataset, we'll need to use the **dzipois()** distribution.

Keep in mind, that a zero-inflated Poisson regression model always takes the general form

$$\begin{aligned} y_i &\sim \text{ZIPoisson}(p_i, \lambda_i) \\ \text{logit}(p_i) &\sim \alpha_p + \beta_p x_i \\ \log(\lambda_i) &\sim \alpha_\lambda + \beta_\lambda x_i \end{aligned}$$

and in this type of model, we have two linear models and two link functions (one for each process in the ZIPoisson). We're allowed to use different predictors in the two models, so there are a lot more possible combinations than in a typical regression problem like in the previous chapters. Thus, model exploration this could go on for a really long time, so I'll just try out a couple since we weren't asked to create a specific one.

The first model will be

$$\begin{aligned} \text{fish} &\sim \text{ZIPoisson}(p, \mu) \\ \text{logit}(p) &\sim \alpha_p + \beta_{pp}\text{persons} + \beta_{pc}\text{child} \\ \log(\mu) &\sim \alpha_\mu + \beta_{\mu p}\text{persons} + \beta_{\mu c}\text{child} + \beta_{\mu h}\log(\text{hours}) \end{aligned}$$

```
lmod11h1_1 <- map2stan(alist(
  fish_caught ~ dzipois(p, mu),
```

```
  logit(p) <- ap + bpp*persons + bpc*child,
  log(mu) <- am + bmp*persons + bmc*child + bmh*hours,
  c(ap,am) ~ dnorm(0,10),
  c(bpp, bpc, bmp, bmc, bmh) ~ dnorm(0,1)
), data = f, iter = 10000, chains = 1, cores = 1)
```

```
## Warning: There were 1 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
## Computing WAIC
```

```
## Constructing posterior predictions
```

In the second model, I'll include the **livebait** variable. Thus, the second model will be:

$$\begin{aligned} \text{fish} &\sim \text{ZIPoisson}(p, \mu) \\ \text{logit}(p) &\sim \alpha_p + \beta_{pp}\text{persons} + \beta_{pc}\text{child} + \beta_{pl}\text{livebait} \\ \log(\mu) &\sim \alpha_\mu + \beta_{\mu p}\text{persons} + \beta_{\mu p} + \beta_{\mu c}\text{child} + \beta_{\mu l}\text{livebait} + \beta_{\mu h}\log(\text{hours}) \end{aligned}$$

```
lmod11h1_2 <- map2stan(alist(
  fish_caught ~ dzipois(p, mu),
  logit(p) <- ap + bpp*persons + bpc*child + bpl*livebait,
  log(mu) <- am + bmp*persons + bmc*child + bml*livebait + bmh*hours,
  c(ap,am) ~ dnorm(0,10),
  c(bpp, bpc, bmp, bmc, bmh, bpl, bml) ~ dnorm(0,1)
), data = f, iter = 10000, chains = 1, cores = 1)
```

```
## Warning: There were 1 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
## Computing WAIC
```

```
## Constructing posterior predictions
```

Now we can compare the two models.

```
precis(lmod11h1_1)
```

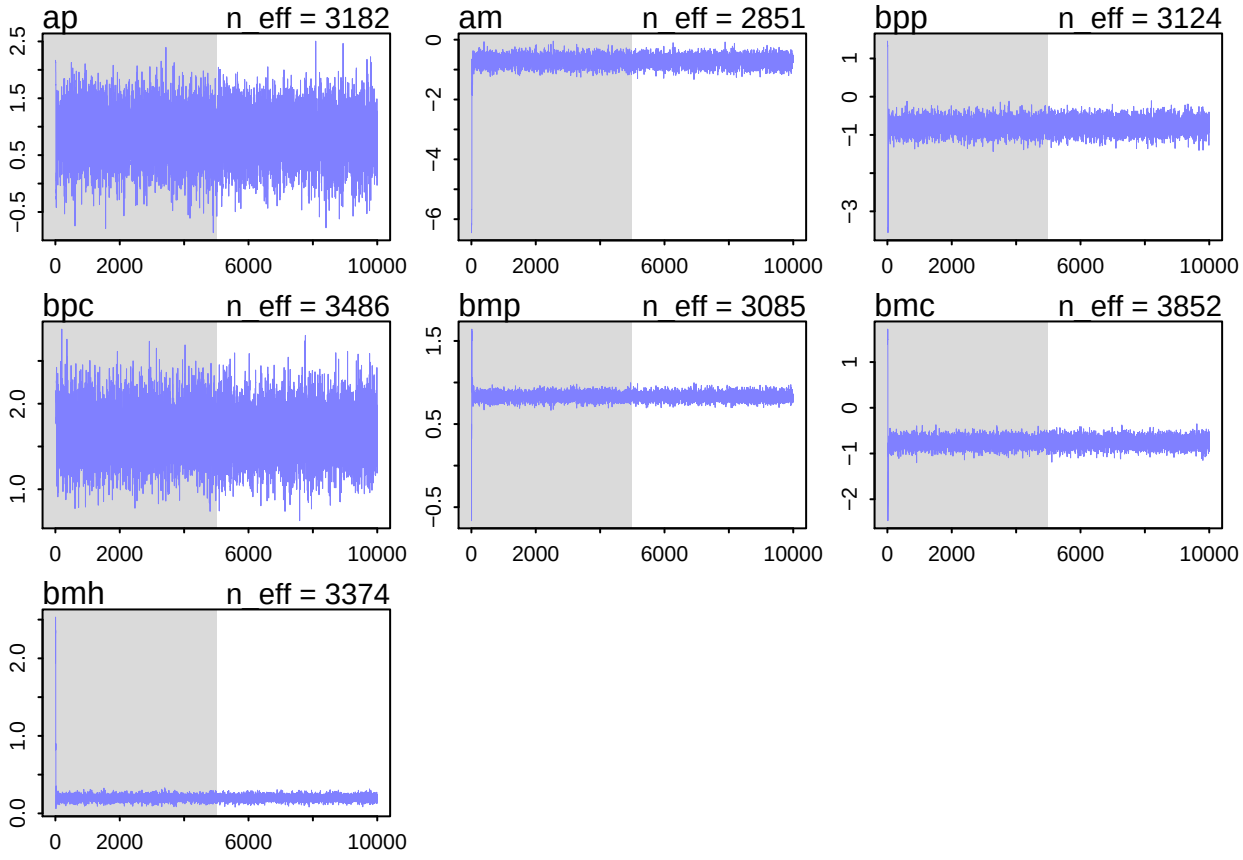
```
##      Mean StdDev lower 0.89 upper 0.89 n_eff Rhat
## ap   0.85   0.44       0.18       1.58  3182    1
## am  -0.71   0.17      -0.98      -0.43  2851    1
## bpp -0.76   0.19      -1.04      -0.45  3124    1
## bpc  1.65   0.29       1.19       2.11  3486    1
## bmp  0.83   0.04       0.76       0.90  3085    1
## bmc -0.75   0.11      -0.93      -0.59  3852    1
## bmh  0.20   0.03       0.14       0.25  3374    1
```

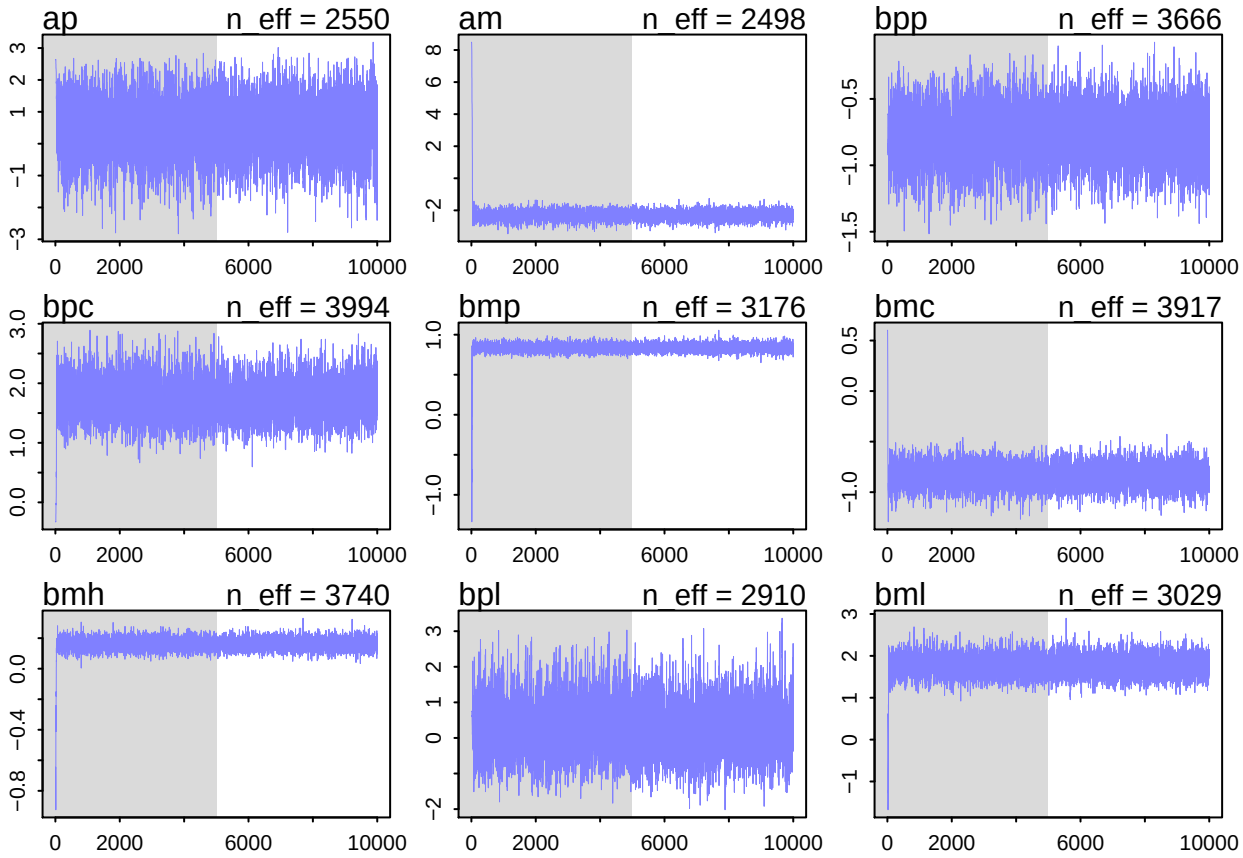
```
precis(lmod11h1_2)
```

```
##      Mean StdDev lower 0.89 upper 0.89 n_eff Rhat
## ap   0.52   0.84      -0.80       1.88  2550    1
## am  -2.30   0.28      -2.73      -1.84  2498    1
## bpp -0.78   0.19      -1.09      -0.48  3666    1
## bpc  1.73   0.30       1.25       2.19  3994    1
## bmp  0.84   0.05       0.77       0.91  3176    1
```

```
## bmc -0.84    0.11      -1.01      -0.67  3917     1
## bmh  0.16    0.03       0.11       0.22  3740     1
## bpl  0.24    0.72      -0.88       1.37  2910     1
## bml  1.75    0.24       1.37       2.12  3029     1
```

```
plot(lmod11h1_1)
plot(lmod11h1_2)
```






```
compare(lmod11h1_1, lmod11h1_2)
```

```
##               WAIC pWAIC dWAIC weight     SE  dSE
## lmod11h1_2 1579.2 101.2   0.0      1 345.90   NA
## lmod11h1_1 1684.0 101.4 104.8      0 373.97 52.7
```

The Rhat values and n_eff values are pretty good for both models, but we can see that the traceplots for the second model are a bit better, and the second model took 100% of the Akaike weight. Thus we can conclude that our second model was the "better" model.

This question doesn't actually ask us to do anything aside from creating the models, but it would still be interesting to explore our model a bit to see how zero-inflated problems are a bit different than the models we've used previously.

```
f$fish_caught
```

```
##   [1]   0   0   0   0   1   0   0   0   0   1   0   0   1   2   0   1   0
##  [18]   0   1   0   1   5   0   3  30   0  13   0   0   0   0   0  11   5
##  [35]   0   1   1   7   0  14   0  32   0   1   0   0   0   1   5   0   1
##  [52]   0  22   0  15   0   0   0   5   4   2   0   2  32   0   0   1   0
##  [69]   0   0   7   0   0   0   0   0   0   0   0   2   3   1   5   0   2
##  [86]   1   0   1 149   0   1   0   0   1   0   0   0   2   2  29   3   0
## [103]   0   5   0   0   0   0   0   1   7   1   0   2   0   2   0   0   0
## [120]   1   0   0   0   0   0   3   4   3   3   8   2   1   6   0   0   5
## [137]   3  31   0   2   0   0   0   0   0   0   6   9   0   0   0   0   0
## [154]   2  15   1   2   3   0  65   5   0   0   0   0   1   8   0   0   0
## [171]   2   4   5   9   0   0   0   0  21   0   6   0   0   0   0  16   0
## [188]   0   4   2  10   0   0   0   2   1   3   0   0  21   0   0   2   0
## [205]   3   0  38   0   0   0   1   3   0   1   0   0   0   0   5   0   0
```

```
## [222]   2   0   0   0   1   4   0   0   2   3   0   0   0   0   1   2   0
## [239]   6   4   1   1   0   1   0   0   0   0   0   0
```

As you can see, there are a LOT of zeroes. But, using our model from **map2stan()**, we can simulate some counterfactual data just with using the **link()** function. Let's see how many fish we would expect a group of 4 adults to catch if they went fishing for 3 hours using livebait, since this is a pretty typical situation in real life. We have to keep in mind, though, that we applied a logarithmic transformation to the **hours** variable, so we have to apply this same transformation to the 3 hours.

```
new_data <- list(hours = log(3), persons = 4, child = 0, livebait = 1)
counterfactual_data <- link(1mod11h1_2, new_data)
```

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

```
p <- counterfactual_data$p
mu <- counterfactual_data$mu
```

Now, according to Wikipedia, the mean of the Zero-Inflated Poisson Distribution is

$$(1-p)\mu$$

Thus we have

```
mean((1-p)*mu)
```

```
## [1] 17.84252
```

Based on this simulation, we can see that a group of 4 adults fishing for 3 hours with live bait would be expected to catch 17.8425159 fish on average (this accounts for the zero-inflation). This sounds pretty reasonable, especially when we take a look at the distribution of the **fish_caught** variable.

```
f[order(-f$fish_caught)[1:5],]
```

```
##     fish_caught livebait camper persons child    hours
## 89          149        1      1       4     0 3.572121
## 160          65        1      1       4     0 1.362258
## 207          38        1      1       4     0 1.189367
## 42           32        1      1       4     0 2.355652
## 64           32        1      1       4     0 2.547960
```

We can see that in the data, there are quite a few groups of 4 adult fishers with livebait who caught quite a lot more than our estimate – but keep in mind that our data was inflated with zeroes, which pulled down the expected mean. (It's not terrrrrrrribly far off though; you can see there's one group who was there for ~2.5h and caught 32 fish, so we're at least on the right order of magnitude. This might suggest that there could've been better models though.)

References

1. https://en.wikipedia.org/wiki/Zero-inflated_model#Zero-inflated_Poisson