

EEEB UN3005/GR5005

Homework - Week 13 - Due 30 Apr 2019

Xun Zhao, xz2827

Homework Instructions: Complete this assignment by writing code in the code chunks provided. If required, provide written explanations below the relevant code chunks. Replace “USE YOUR NAME HERE” with your name in the document header. When complete, knit this document within RStudio to generate a pdf. Please review the resulting pdf to ensure that all content relevant for grading (i.e., code, code output, and written explanations) appears in the document. Rename your pdf document according to the following format: hw_week_13_firstname_lastname.pdf. Upload this final homework document to CourseWorks by 5 pm on the due date.

This week’s homework problems will use the snow goose color morph dataset that was introduced at the end of last week’s lecture. The following code will load this dataset into your R environment as a data frame called `geese`:

```
# Load in snow goose data in aggregated binomial format
geese <- data.frame(
  blue_geese = as.integer(c(215, 84, 7)),
  total_geese = as.integer(c(500, 300, 25)),
  study_site = as.integer(c(1, 2, 3))
)
```

Problem 1 (4 points)

Use `map()` to fit a binomial generalized linear model, with `study_site` as a predictor variable and `blue_geese` as the outcome variable. Rather than constructing dummy variables to represent site affiliation, use the existing `study_site` variable as an index variable to create a vector of intercept parameters. This model should be analogous to the model that was demonstrated during the week 12 lecture. However, with this model formulation, you’ll generate an intercept value that directly corresponds to each study site’s log-odds of a goose belonging to the blue morph.

After fitting the model, use `precis()` to report the 97% PIs of the fit model parameters. Use posterior samples from the model and `dens()` to visualize the implied probability of success (i.e., implied probability of a goose being blue) values for each of the three study sites.

```
model.1 = map(
  alist(
    blue_geese ~ dbinom(total_geese, p),
    logit(p) <- a[study_site],
    a[study_site] ~ dnorm(0, 5)),
  data = geese)
```

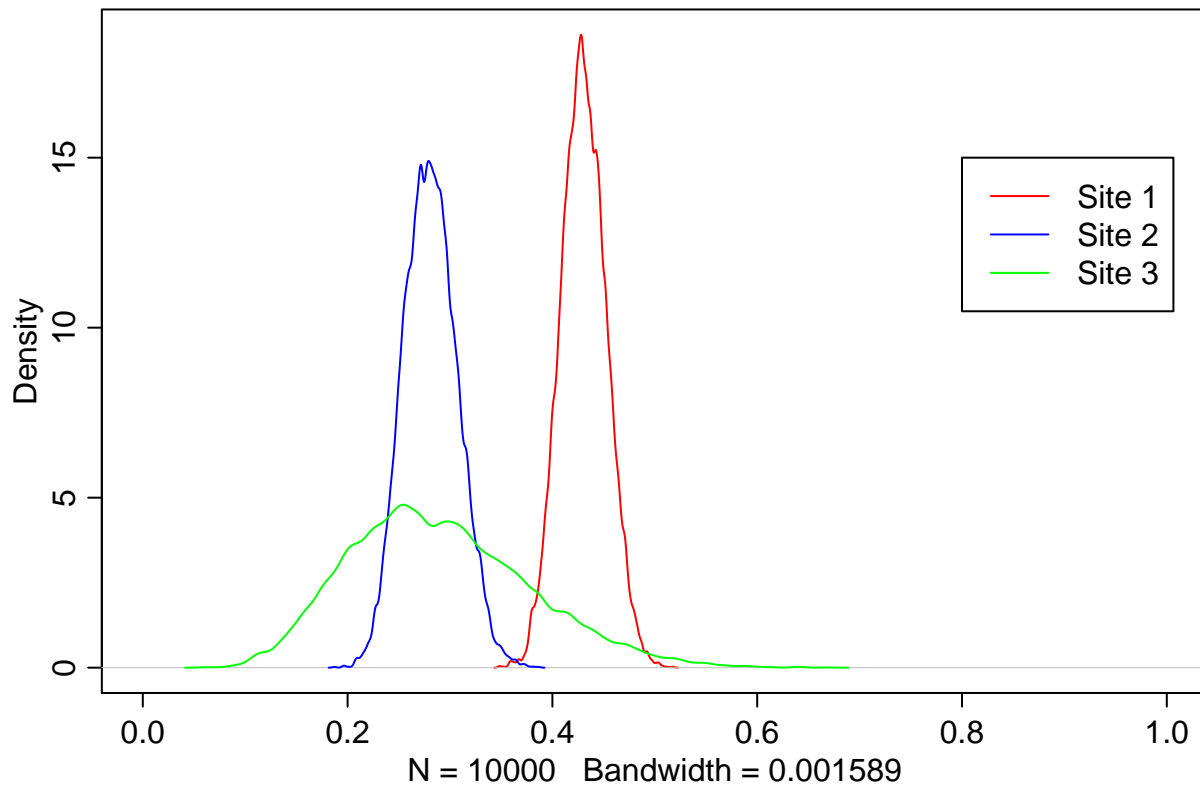
```

data = geese)
precis(model.1, depth = 2, prob = 0.97)

##           Mean StdDev 1.5% 98.5%
## a[1] -0.28    0.09 -0.48 -0.09
## a[2] -0.94    0.13 -1.22 -0.66
## a[3] -0.94    0.44 -1.90  0.02

sample.1 = extract.samples(model.1, 10000)
dens(logistic(sample.1$a[,1]), col = 'red', xlim = c(0, 1))
dens(logistic(sample.1$a[,2]), add = TRUE, col = 'blue')
dens(logistic(sample.1$a[,3]), add = TRUE, col = 'green')
legend(0.8, 15, legend = c('Site 1', 'Site 2', 'Site 3'),
      col = c('red', 'blue', 'green'), lty = 1)

```



Problem 2 (6 points)

Now, formulate and fit this same model as a multilevel binomial generalized linear model. Remember, you'll have to use `map2stan()` in order to successfully fit a multilevel model. Don't worry about any warning messages you may receive regarding divergent iterations during sampling.

After fitting the model, visualize the MCMC chain(s) using a trace plot function, and use `precis()` to report the 97% HPDIs of the fit model parameters. Use posterior samples from the model and `dens()` to visualize the implied probability of success (i.e., implied probability of a goose being blue) values for each of the three study sites.

What differences, if any, do you see between the posterior probability of success values for each site with this model compared to the model fit in Problem 1? Referencing your `precis()` output for both models (which show raw model estimates on the log-odds scale) may also help with interpretation.

```
model.2 = map2stan(
  alist(
    blue_geese ~ dbinom(total_geese, p),
    logit(p) <- a[study_site],
    a[study_site] ~ dnorm(mu, sigma),
    mu ~ dnorm(0, 1),
    sigma ~ dcauchy(0, 1)),
  chains = 4,
  data = geese,
  iter = 4000)
```

```
## Warning: There were 11 divergent transitions after warmup. Increasing adapt_delta above
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

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

## Warning: There were 1 divergent transitions after warmup. Increasing adapt_delta above
## 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

## Aggregated binomial counts detected. Splitting to 0/1 outcome for WAIC calculation.

## Warning in map2stan(alist(blue_geese ~ dbinom(total_geese, p), logit(p) <- a[study_si
## Check the chains (trace plots, n_eff, Rhat) carefully to ensure they are valid.
```

```
precis(model.2, prob = 0.97, depth = 2)
```

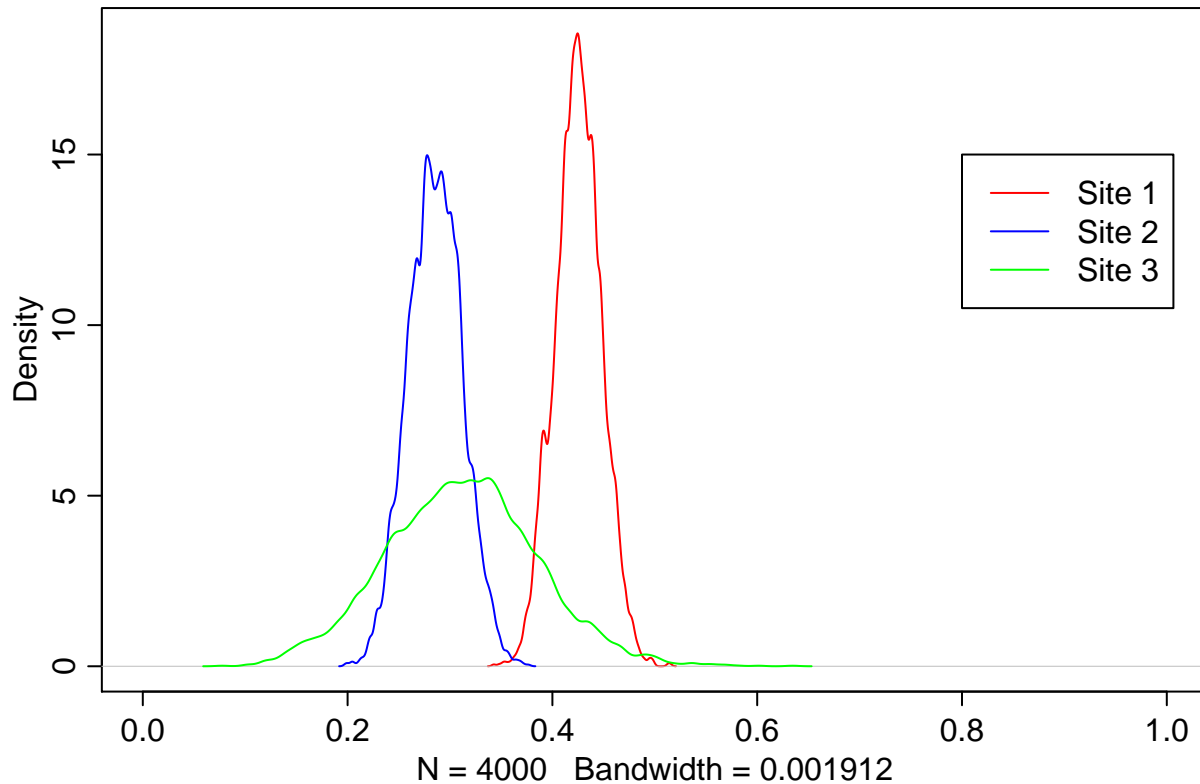
```
## Warning in precis(model.2, prob = 0.97, depth = 2): There were 11 divergent iteration
## Check the chains (trace plots, n_eff, Rhat) carefully to ensure they are valid.
```

```
##           Mean StdDev lower 0.97 upper 0.97 n_eff Rhat
## a[1]  -0.30   0.09    -0.51   -0.11  6398    1
## a[2]  -0.92   0.13    -1.20   -0.64  6293    1
## a[3]  -0.82   0.36    -1.69   -0.08  5334    1
## mu    -0.58   0.39    -1.51    0.40  3757    1
## sigma  0.65   0.49     0.10    1.85  2766    1
```

```

sample.2 = extract.samples(model.2, 4000)
dens(logistic(sample.2$a[,1]), col = 'red', xlim = c(0, 1))
dens(logistic(sample.2$a[,2]), add = TRUE, col = 'blue')
dens(logistic(sample.2$a[,3]), add = TRUE, col = 'green')
legend(0.8, 15, legend = c('Site 1', 'Site 2', 'Site 3'),
      col = c('red', 'blue', 'green'), lty = 1)

```



Answer:

In model 1, the estimates of a_3 is smaller than that in model 2. So the estimates probability is smaller, too. However, in model 1, the standard deviation of a_3 is larger.

Other estimates about a_1 and a_2 keep the same.