EEEB UN3005/GR5005
Homework - Week 12 - Due 23 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_12_firstname_lastname.pdf. Upload this final homework document to CourseWorks by 5 pm on the due date.

This week's homework problems will ask you to think all the way back to your week 04 homework assignment. To briefly summarize, you were asked to imagine studying a bacterial pathogen that infects small mammals. Hypothetical pilot research efforts found 9 infected individuals out of 20 animals sampled in total.

All the problems in this homework assignment will ask you to work with this data scenario. The primary challenge should not be the specification and fitting of the models; they'll each contain only one parameter, the intercept value. Rather, I'm mainly asking you to think about how to represent the data in different formats that are all valid expressions of binomial data.

Problem 1 (3 points)

First, represent the data scenario (9 infected animals out of 20 animals sampled) using an *aggregated binomial* data format. For this problem, your data should be contained in a data frame with only one row and two columns. One column should contain the number of infected individuals. The other column should contain the total number of samples.

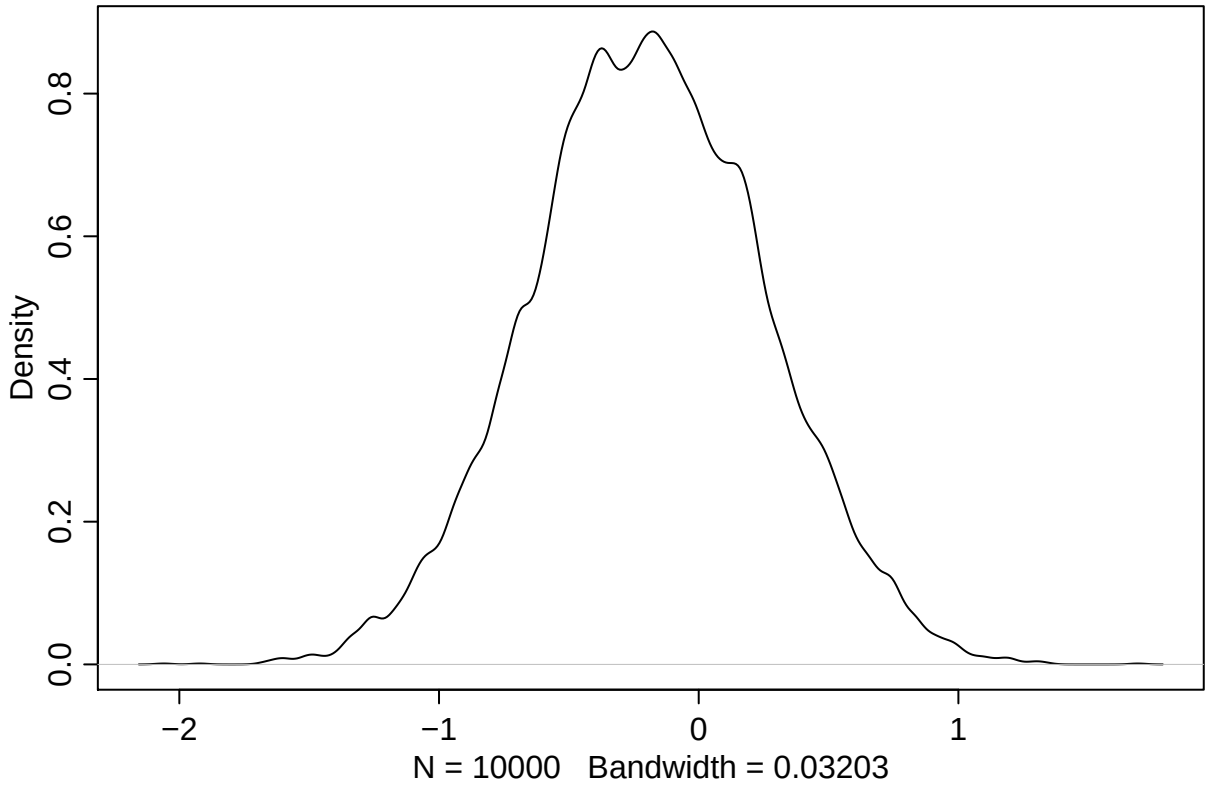
Use `map()` to fit a binomial generalized linear model to this data (an intercept-only model with no predictor variables). Explicitly specify a start value of -5 for your intercept parameter. After fitting, use `precis()` to report the 97% PI of the fit intercept parameter. Further, use posterior samples from the model and `dens()` to visualize the posterior distribution of the intercept parameter **and** its implied probability of success (i.e., probability of infection) values.

```
d = data.frame(infect = 9, total = 20)
model = map(
            alist(
                  infect ~ dbinom(total, p),
                  logit(p) <- a,
```

```
                    a ~ dnorm(0, 5)),
              start = list(a = -5),
              data = d)
precis(model, 0.97)
```

```
##   Mean StdDev  5.5% 94.5%
## a -0.2   0.45 -0.91  0.52
```

```
sample = extract.samples(model, 10000)
dens(sample$a)
```



Answer:

The implied probability is the value of reverse link funtion of a. So it is `p = logistic(a) =` 0.4513448.

Problem 2 (3 points)

Now refit the same model, but this time construct your data such that each row represents a single binomial trial. In other words, this is disaggregated binomial data, leading to what was termed in lecture *logistic regression*. For this data format, you will need a data frame with only one column. Since each row will represent a single binomial trial, the `size` argument within your model's `dbinom()` call will simply be equal to 1. After fitting the model, report the 97% PI of the fit intercept parameter to confirm you get identical posterior inference to Problem 1.

```
d.2 = data.frame(infect = c(rep(1, 9), rep(0, 11)))
model.2 = map(
              alist(
                    infect ~ dbinom(1, p),
                    logit(p) <- a,
                    a ~ dnorm(0, 5)),
              start = list(a = -5),
              data = d.2)
precis(model.2, 0.97)
```

```
##   Mean StdDev  5.5% 94.5%
## a -0.2   0.45 -0.91  0.52
```

Problem 3 (4 points)

Refit the same model again (using either data format), but this time use `map2stan()`, specifying four MCMC chains. In addition, use a `dunif()` prior for your model's intercept parameter that encodes the assumption that probability of success (i.e., probability of infection) values > 0.5 are impossible. Remember, since the binomial GLM uses a link function to relate the probability of success value (which must be between 0 and 1) to the fit model parameters, you'll have to think about what values your intercept parameter should be constrained to...

After fitting the model, use `precis()` to report the 97% HPDI of the fit intercept parameter. Further, use posterior samples from the model and `dens()` to visualize the intercept parameter and its implied probability of success (i.e., probability of infection) values.

```
model.3 = map2stan(
                   alist(
                         infect ~ dbinom(1, p),
                         logit(p) <- a,
                         a ~ dunif(-2 ^ 32, 0)),
                   start = list(a = -5),
                   data = d.2,
                   chains = 4)
```

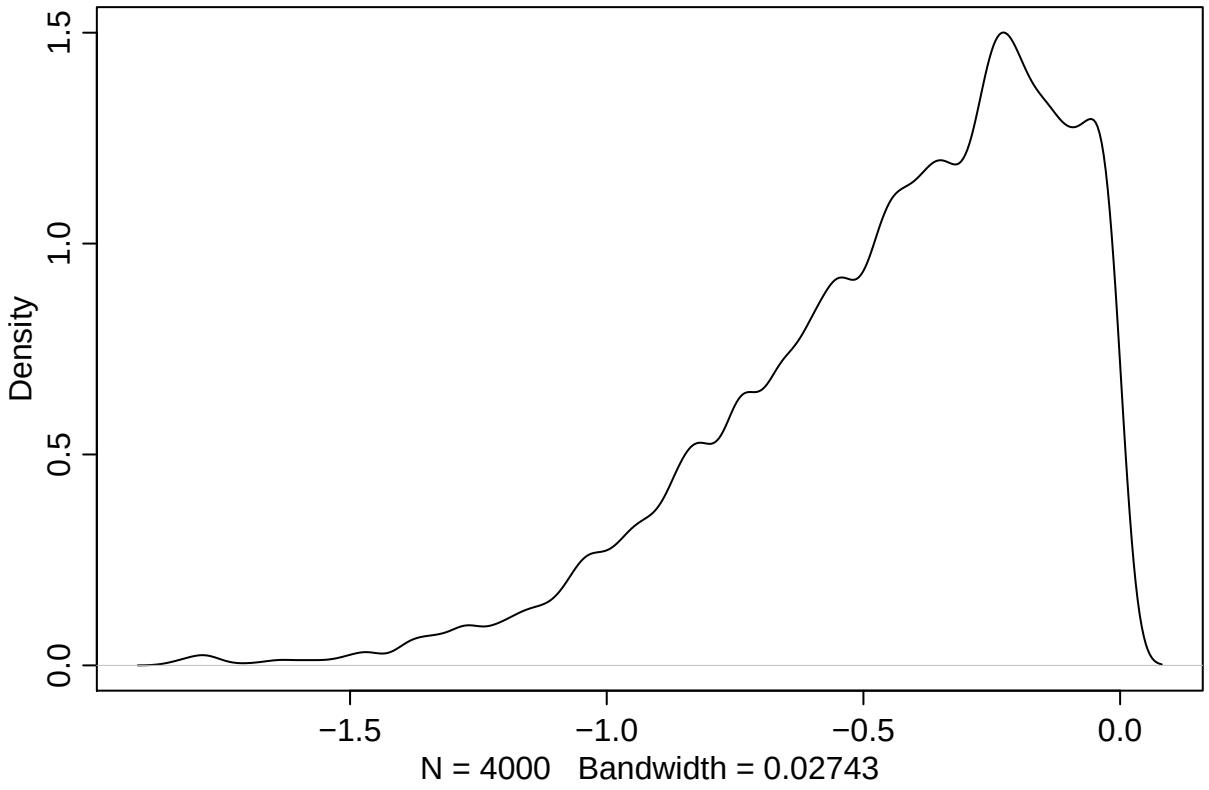
```
## Computing WAIC
```

```
## Constructing posterior predictions
```

```
precis(model.3, 0.97)
```

```
##    Mean StdDev lower 0.89 upper 0.89 n_eff Rhat
## a -0.44   0.32      -0.86          0  1030 1.01
```

```
sample.3 = extract.samples(model.3, 10000)
dens(sample.3$a)
```




```
mean(logistic(sample.3$a), na.rm = TRUE)
```

```
## [1] 0.3954273
```