

Chapter 10 Questions

Melissa Van Busse

July 20, 2018

Points for chapter 10: 2 + 8 + 4 = 14 total points.

Easy Questions (2 points total)

10E1

```
log(0.35 / (1 - 0.35))
```

```
## [1] -0.6190392
```

10E2

```
library(rethinking)  
inv_logit(3.2)
```

```
## [1] 0.9608343
```

10E3

```
q10e3 <- exp(1.7)  
q10e3
```

```
## [1] 5.473947
```

This implies that for every one unit increase in the predictor variable, the odds of the event are multiplied by 5.4739474.

10E4

If some observations came from longer periods than others, then you would use an offset so account for the fact that we expect counts from larger periods to be bigger.

Medium Questions (8 points total)

10M2 (2 points)

```
increase <- exp(1.7)
```

It means that when the predictor variable increases by one unit, the expected count will increase by 5.4739474.

10M3 (2 points)

We usually use a logit link for a binomial generalized linear model. This is because the logit allows us to map continuous values to a probability bounded between 0 and 1.

10M4 (2 points)

We usually use a log function for a Poisson generalized linear model. This is because the log allows us to map continuous values to values that are strictly positive.

10M5 (2 points)

If you were to use a logit function for a Poisson GLM, then λ would have to lie between 0 and 1. A real world situation where this might be used would be if you knew that the true rate was extremely small

10M6 (2 points)

The 2 distributions have the same constraints since Poisson is basically just a simplified version of the binomial. The two constraints are:

- Discrete, binary outcomes
- Constant expected values

Hard Questions (4 points total)

10H1 (2 points)

Just copy paste the model from the example in the textbook and use **map()** instead of **map2stan()**.

```
data(chimpanzees)
d <- chimpanzees
d2 <- d
d2$recipient <- NULL
m10.4_map <- map(
  alist(
    pulled_left ~ dbinom(1, p),
    logit(p) <- a[actor] + (bp + bpC*condition)*prosoc_left,
    a[actor] ~ dnorm(0,10),
    bp ~ dnorm(0,10),
    bpC ~ dnorm(0,10)),
  data = d2)
```

10H2 (2 points)

First we need to also have the model from the chapter (so, it's the one from the last question but using **map2stan()** instead of **map()**):

```
m10.4 <- map2stan(
  alist(
    pulled_left ~ dbinom(1, p),
    logit(p) <- a[actor] + (bp + bpC*condition)*prosoc_left,
    a[actor] ~ dnorm(0,10),
    bp ~ dnorm(0,10),
    bpC ~ dnorm(0,10)),
  data = d2)
```

```
## Computing WAIC
```

```
## Constructing posterior predictions
```

Now we can compare the two by looking at the precis for each:

```
precis(m10.4_map)
```

```
## 7 vector or matrix parameters omitted in display. Use depth=2 to show them.
```

```
##      Mean StdDev  5.5% 94.5%
## bp   0.82   0.26  0.40  1.24
## bpC -0.13   0.30 -0.61  0.34
```

```
precis(m10.4)
```

```
## 7 vector or matrix parameters omitted in display. Use depth=2 to show them.
```

```
##      Mean StdDev lower 0.89 upper 0.89 n_eff Rhat
## bp   0.84   0.25     0.45     1.24   595    1
## bpC -0.13   0.30    -0.56     0.39   880    1
```

The main difference is in the coefficient for the second intercept.

Now we can compare their WAIC values:

```
WAIC(m10.4)
```

```
## [1] 529.4909
## attr(,"lppd")
## [1] -256.6132
## attr(,"pWAIC")
## [1] 8.132248
## attr(,"se")
## [1] 19.94233
```

```
WAIC(m10.4_map)
```

```
## Constructing posterior predictions
```

```
## [ 100 / 1000 ]
## [ 200 / 1000 ]
## [ 300 / 1000 ]
## [ 400 / 1000 ]
## [ 500 / 1000 ]
## [ 600 / 1000 ]
## [ 700 / 1000 ]
## [ 800 / 1000 ]
## [ 900 / 1000 ]
## [1000 / 1000 ]
## [1] 564.9634
```

```
## attr("lppd")  
## [1] -260.1199  
## attr("pWAIC")  
## [1] 22.36185  
## attr("se")  
## [1] 17.95934
```

Not sure why this question even asked us to compare their WAIC values, because Richard McElreath built it right into his package that you shouldn't be comparing models computed using different algorithms... anyways, though, the WAIC for the map estimate is larger than the WAIC for the MCMC estimate, which makes sense.