

# glm

Chris Parrish  
June 29, 2016

## Contents

<b>UCB admittance</b>	<b>1</b>
data .....	1
glm .....	2
<b>prosocial</b>	<b>2</b>
data .....	2
glm .....	3
<b>outcome and predictor almost perfectly associated</b>	<b>3</b>
bad fit .....	3
good fit .....	3
stan .....	4

glm

reference: McElreath, Statistical Rethinking, chap 10, p.304

```
library(rethinking)
library(ggplot2)
```

## UCB admittance

data

```
## R code 10.22
data(UCBadmit)
str(UCBadmit)

## 'data.frame':  12 obs. of  5 variables:
## $ dept      : Factor w/ 6 levels "A","B","C","D",...: 1 1 2 2 3 3 4 4 5 5 ...
## $ applicant.gender: Factor w/ 2 levels "female","male": 2 1 2 1 2 1 2 1 2 1 ...
## $ admit      : int  512 89 353 17 120 202 138 131 53 94 ...
## $ reject     : int  313 19 207 8 205 391 279 244 138 299 ...
## $ applications : int  825 108 560 25 325 593 417 375 191 393 ...

d <- UCBadmit
## R code 10.23
d$male <- ifelse( d$applicant.gender=="male" , 1 , 0 )
str(d)

## 'data.frame':  12 obs. of  6 variables:
## $ dept      : Factor w/ 6 levels "A","B","C","D",...: 1 1 2 2 3 3 4 4 5 5 ...
## $ applicant.gender: Factor w/ 2 levels "female","male": 2 1 2 1 2 1 2 1 2 1 ...
## $ admit      : int  512 89 353 17 120 202 138 131 53 94 ...
## $ reject     : int  313 19 207 8 205 391 279 244 138 299 ...
```

```
## $ applications      : int  825 108 560 25 325 593 417 375 191 393 ...
## $ male              : num  1 0 1 0 1 0 1 0 1 0 ...
```

## glm

```
## R code 10.32
m10.7glm <- glm( cbind(admit,reject) ~ 1 , data=d , family=binomial )
m10.6glm <- glm( cbind(admit,reject) ~ male , data=d , family=binomial )
m10.8glm <- glm( cbind(admit,reject) ~ dept , data=d , family=binomial )
m10.9glm <- glm( cbind(admit,reject) ~ male + dept , data=d ,
  family=binomial )
```

```
options(show.signif.stars=FALSE)
summary(m10.9glm)
```

```
##
## Call:
## glm(formula = cbind(admit, reject) ~ male + dept, family = binomial,
##      data = d)
##
## Deviance Residuals:
##      1      2      3      4      5      6      7      8
## -1.2487  3.7189 -0.0560  0.2706  1.2533 -0.9243  0.0826 -0.0858
##      9     10     11     12
##  1.2205 -0.8509 -0.2076  0.2052
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.68192    0.09911   6.880 5.97e-12
## male        -0.09987    0.08085  -1.235  0.217
## deptB       -0.04340    0.10984  -0.395  0.693
## deptC       -1.26260    0.10663 -11.841 < 2e-16
## deptD       -1.29461    0.10582 -12.234 < 2e-16
## deptE       -1.73931    0.12611 -13.792 < 2e-16
## deptF       -3.30648    0.16998 -19.452 < 2e-16
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 877.056  on 11  degrees of freedom
## Residual deviance:  20.204  on  5  degrees of freedom
## AIC: 103.14
##
## Number of Fisher Scoring iterations: 4
```

## prosocial

### data

```
## R code 10.33
data(chimpanzees)
```

## glm

```
m10.4glm <- glm(
  pulled_left ~ as.factor(actor) + prosoc_left * condition - condition ,
  data=chimpanzees , family=binomial )

## R code 10.34
glimmer( pulled_left ~ prosoc_left * condition - condition ,
  data=chimpanzees , family=binomial )

## alist(
##   pulled_left ~ dbinom( 1 , p ),
##   logit(p) <- Intercept +
##     b_prosoc_left*prosoc_left +
##     b_prosoc_left_X_condition*prosoc_left_X_condition,
##   Intercept ~ dnorm(0,10),
##   b_prosoc_left ~ dnorm(0,10),
##   b_prosoc_left_X_condition ~ dnorm(0,10)
## )
```

## outcome and predictor almost perfectly associated

```
## R code 10.35
# outcome and predictor almost perfectly associated
y <- c( rep(0,10) , rep(1,10) )
x <- c( rep(-1,9) , rep(1,11) )
```

## bad fit

```
# fit binomial GLM
m.bad <- glm( y ~ x , data=list(y=y,x=x) , family=binomial )
precis(m.bad)

##           Mean StdDev   5.5%  94.5%
## (Intercept) -9.13 2955.06 -4731.89 4713.63
## x           11.43 2955.06 -4711.33 4734.19
```

## good fit

```
## R code 10.36
m.good <- map(
  alist(
    y ~ dbinom( 1 , p ),
    logit(p) <- a + b*x,
    c(a,b) ~ dnorm(0,10)
  ) , data=list(y=y,x=x) )
precis(m.good)

##           Mean StdDev   5.5%  94.5%
```

```
## a -1.73  2.77 -6.16  2.71
## b  4.02  2.77 -0.42  8.45
```

## stan

```
## R code 10.37
m.good.stan <- map2stan( m.good )

##
## SAMPLING FOR MODEL 'y ~ dbinom(1, p)' NOW (CHAIN 1).
##
## Chain 1, Iteration:    1 / 2000 [  0%] (Warmup)
## Chain 1, Iteration:   200 / 2000 [ 10%] (Warmup)
## Chain 1, Iteration:   400 / 2000 [ 20%] (Warmup)
## Chain 1, Iteration:   600 / 2000 [ 30%] (Warmup)
## Chain 1, Iteration:   800 / 2000 [ 40%] (Warmup)
## Chain 1, Iteration:  1000 / 2000 [ 50%] (Warmup)
## Chain 1, Iteration:  1001 / 2000 [ 50%] (Sampling)
## Chain 1, Iteration:  1200 / 2000 [ 60%] (Sampling)
## Chain 1, Iteration:  1400 / 2000 [ 70%] (Sampling)
## Chain 1, Iteration:  1600 / 2000 [ 80%] (Sampling)
## Chain 1, Iteration:  1800 / 2000 [ 90%] (Sampling)
## Chain 1, Iteration:  2000 / 2000 [100%] (Sampling)
## Elapsed Time: 0.032259 seconds (Warm-up)
##                0.029076 seconds (Sampling)
##                0.061335 seconds (Total)
##
##
## SAMPLING FOR MODEL 'y ~ dbinom(1, p)' NOW (CHAIN 1).
## WARNING: No variance estimation is
##           performed for num_warmup < 20
##
##
## Chain 1, Iteration: 1 / 1 [100%] (Sampling)
## Elapsed Time: 3e-06 seconds (Warm-up)
##                4.7e-05 seconds (Sampling)
##                5e-05 seconds (Total)

## Computing WAIC

## Constructing posterior predictions

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

pairs(m.good.stan)
```

