

## R codes for assignment 6

Wen-Shiuan, Liang

12/31/2017

```
rm(list=ls())                                # clear memory
library(MASS)
library(rethinking)

## Loading required package: rstan
## Loading required package: ggplot2
## Loading required package: StanHeaders
## rstan (Version 2.17.2, 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)
library(rstan)
```

### Question 1

(a) Fit the model above to the eagles data, using both map and map2stan. Is the quadratic approximation okay?

```
data(eagles)
d = eagles
d$pp <- ifelse( d$P=="L" , 1 , 0 )
d$v <- ifelse( d$V=="L" , 1 , 0 )
d$aa <- ifelse( d$A=="A" , 1 , 0 )
m1 <- map(
  alist(
    y ~ dbinom( n , p) ,
    logit(p) <- a + bP*pp + bV*v + bA*aa ,
    a ~ dnorm(0,10) ,
    bP ~ dnorm(0,5) ,
    bV ~ dnorm(0,5) ,
    bA ~ dnorm(0,5)
  ) ,
```

```

data=d )
precis(m1)

##      Mean StdDev  5.5% 94.5%
## a    0.59   0.66 -0.47  1.65
## bP   4.24   0.90  2.81  5.67
## bV  -4.59   0.96 -6.13 -3.06
## bA   1.08   0.53  0.23  1.93

d2 = d
d2$recipient = NULL
m2 <- map2stan(m1,d2,iter=10000 , warmup=1000,chain=2)

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

precis(m2)

##      Mean StdDev lower 0.89 upper 0.89 n_eff Rhat
## a    0.64   0.70   -0.52   1.70  8405    1
## bP   4.64   1.01    3.03    6.11  6763    1
## bV  -5.04   1.07   -6.63   -3.31  6293    1
## bA   1.14   0.55    0.29    2.02  9749    1

coefstab(m1,m2)

##      m1      m2
## a    0.59    0.64
## bP   4.24    4.64
## bV  -4.59   -5.04
## bA   1.08    1.14
## nobs      8      8

```

*The quadratic results and that of map2stan have similar effect, telling that quadratic approximation are okay.*

*The coefficient shows that there would be higher chance to succeed if the pirate eagle is big and adult. While if the victim eagle is big, the probability of success would decrease.*

**(b) Now interpret the estimates. If the quadratic approximation turned out okay, then it's okay to use map estimates. Otherwise stick to map2stan estimates. Then plot the posterior predictions. Compute and display both. What different information does each type of posterior prediction provide?**

**(1) the predicted probability of success and its 89% interval for each row(i) in the data**

```

d.pred <- data.frame(
  pp = c(1,1,1,1,0,0,0,0),
  v = c(1,0,1,0,1,0,1,0),
  aa = c(1,1,0,0,1,1,0,0)
)

```

```

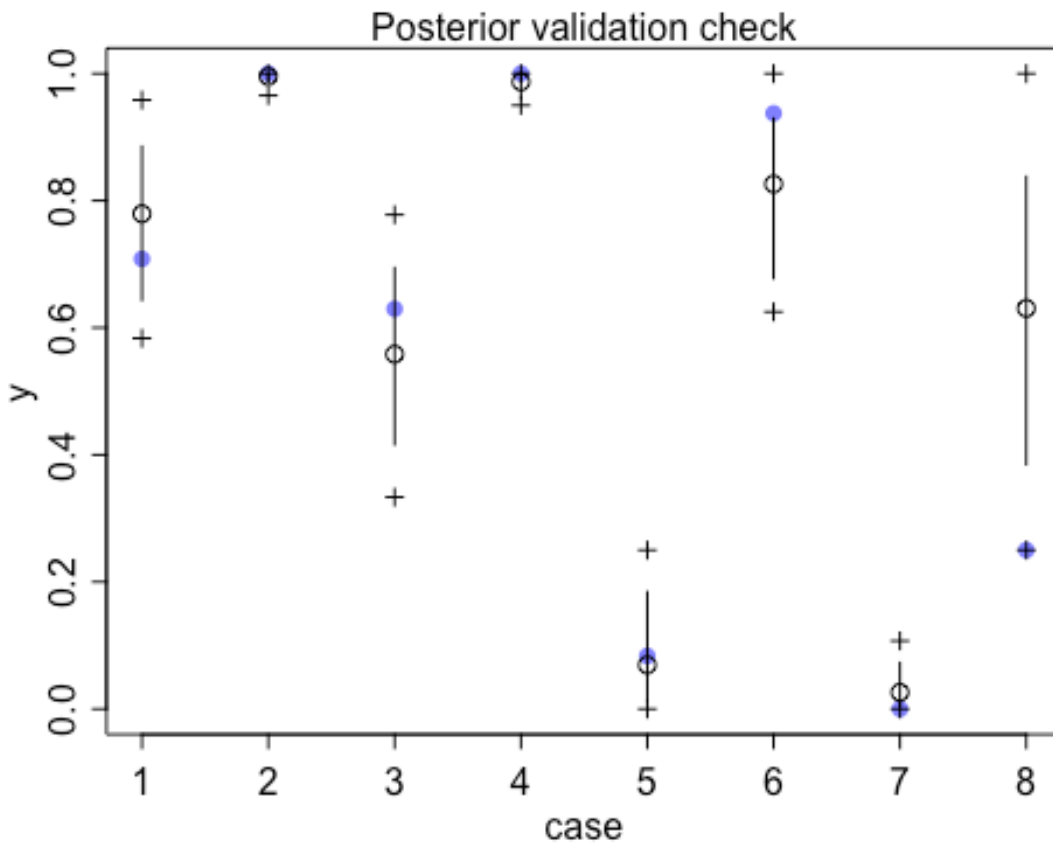
eagles.ensemble <- ensemble( m1, data=d.pred )
pred.p <- apply( eagles.ensemble$link , 2 , mean )
(pred.p.PI <- apply( eagles.ensemble$link , 2 , PI ))

##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## 5%  0.6459127 0.9863000 0.4146745 0.9595069 0.01358746 0.6746740
## 94% 0.8858361 0.9994735 0.6994043 0.9984995 0.18490664 0.9312339
##           [,7]      [,8]
## 5%  0.004184606 0.3832565
## 94% 0.070619553 0.8341335

## R code 10.27
postcheck( m1 , n=1e4 )

## [ 1000 / 10000 ]
[ 2000 / 10000 ]
[ 3000 / 10000 ]
[ 4000 / 10000 ]
[ 5000 / 10000 ]
[ 6000 / 10000 ]
[ 7000 / 10000 ]
[ 8000 / 10000 ]
[ 9000 / 10000 ]
[ 10000 / 10000 ]
## [ 1000 / 10000 ]
[ 2000 / 10000 ]
[ 3000 / 10000 ]
[ 4000 / 10000 ]
[ 5000 / 10000 ]
[ 6000 / 10000 ]
[ 7000 / 10000 ]
[ 8000 / 10000 ]
[ 9000 / 10000 ]
[ 10000 / 10000 ]

```



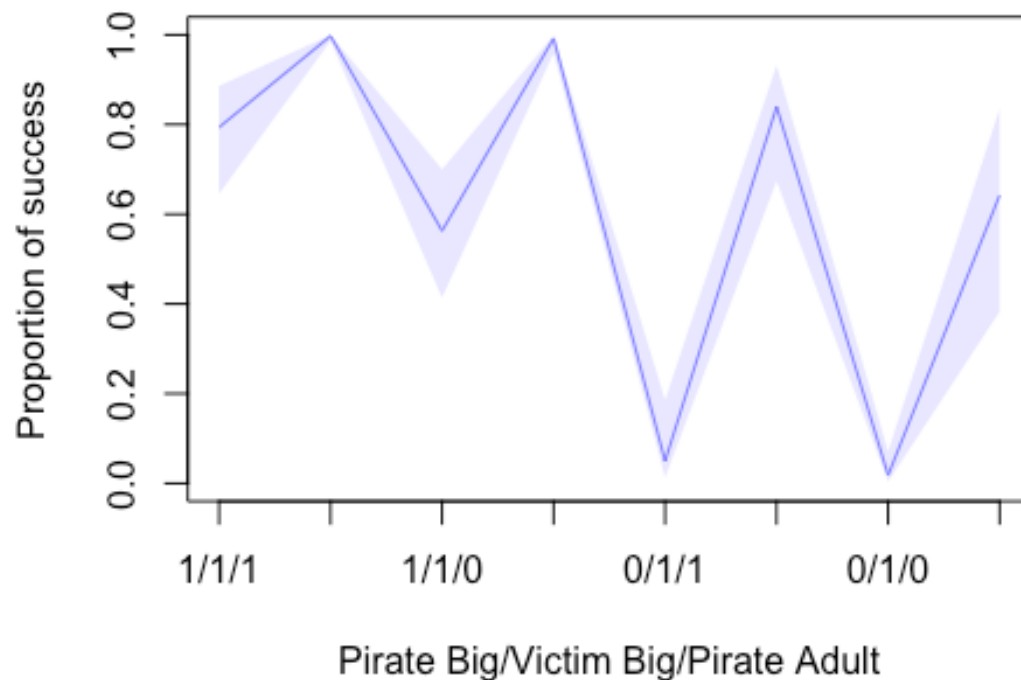
##

(2) the predicted success count and its 89% interval.

```
eagles.med <- apply( eagles.ensemble$link , 2 , median )
eagles.PI <- apply( eagles.ensemble$link , 2 , PI )

plot( 0 , 0 , type="n" , xlab="Pirate Big/Victim Big/Pirate Adult" ,
      ylab="Proportion of success" , ylim=c(0,1) , xaxt="n" ,
      xlim=c(1,8) )
axis( 1 , at=1:8 , labels=c("1/1/1","1/0/1","1/1/0","1/0/0","0/1/1","0/0/1","0/1/0","0/0/0") )

lines( 1:8 , eagles.med , col=range2 )
shade( eagles.PI , 1:8 , col=col.alpha(range2,0.2) )
```



```
#The counts of 89%PI
x=as.numeric(d$n)
counts=data.frame()
for (i in x){
  j=which(x==i)
  counts=rbind(counts, i*pred.p.PI[,j])
}
colnames(counts)=c('5%', '94%')
counts
```

```
##           5%           94%
## 1 15.5019058 21.260067
## 2 28.6027009 28.984730
## 3 11.1962115 18.883916
## 4 19.1901386 19.969991
## 5  0.1630496  2.218880
## 6 10.7947835 14.899743
## 7  0.1171690  1.977347
## 8  1.5330260  3.336534
```

for plot (1) it shows the posterior prediction of probability of success in pirating. For example, the next success rate of pirating for  $P=1, V=1, A=1$ . On the other hand, plot (2) shows the posterior predictions of success counts for current  $n$  of each type of combination. For example, after 24 trials, what would be the total success numbers of  $P=1, V=1, A=1$ .

**(c) Now try to improve the model. Consider an interaction between the pirate's size and age (immature or adult). Compare this model to the previous one, using WAIC to interpret.**

```
m1.2 <- map(
  alist(
    y ~ dbinom( n , p) ,
    logit(p) <- a + bP*pp + bV*v+ bA*aa+bPA*pp*aa ,
    a ~ dnorm(0,10) ,
    bP ~ dnorm(0,5) ,
    bV ~ dnorm(0,5) ,
    bA ~ dnorm(0,5) ,
    bPA ~ dnorm(0,5)
  ) ,
  data=d )
precis(m1.2)
```

```
##      Mean StdDev  5.5% 94.5%
## a    -0.68  0.95 -2.20  0.84
## bP    5.89  1.30  3.81  7.97
## bV   -4.72  1.03 -6.37 -3.07
## bA    3.09  1.15  1.25  4.93
## bPA  -2.65  1.28 -4.70 -0.61
```

*#Try Stan*

```
m1.2.2 <- map2stan(
  alist(
    y ~ dbinom( n , p) ,
    logit(p) <- a + bP*pp + bV*v+ bA*aa+bPA*pp*aa ,
    a ~ dnorm(0,10) ,
    bP ~ dnorm(0,5) ,
    bV ~ dnorm(0,5) ,
    bA ~ dnorm(0,5) ,
    bPA ~ dnorm(0,5)
  ) ,
  data=d, iter=10000 , warmup=1000, chain=2)
```

## Constructing posterior predictions

```
## [ 1800 / 18000 ]
[ 3600 / 18000 ]
[ 5400 / 18000 ]
[ 7200 / 18000 ]
[ 9000 / 18000 ]
[ 10800 / 18000 ]
```

```
[ 12600 / 18000 ]
[ 14400 / 18000 ]
[ 16200 / 18000 ]
[ 18000 / 18000 ]
```

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

```
precis(m1.2.2)
```

```
##      Mean StdDev lower 0.89 upper 0.89 n_eff Rhat
## a    -0.81   1.06    -2.45    0.91  5085    1
## bP    6.60   1.44     4.29    8.89  4560    1
## bV   -5.28   1.17    -6.97   -3.32  6799    1
## bA    3.47   1.27     1.43    5.42  4351    1
## bPA   -3.00   1.39    -5.22   -0.82  4522    1
```

```
compare(m1,m1.2)
```

```
##      WAIC pWAIC dWAIC weight    SE dSE
## m1.2  93.9   4.2   0.0   0.91 11.69  NA
## m1    98.6   3.7   4.7   0.09 12.39  4.13
```

```
coeftab(m1,m1.2,m1.2.2)
```

```
##      m1      m1.2      m1.2.2
## a      0.59    -0.68    -0.81
## bP      4.24     5.89     6.60
## bV     -4.59    -4.72    -5.28
## bA      1.08     3.09     3.47
## bPA      NA    -2.65    -3.00
## nobs      8       8       8
```

```
compare(m2,m1.2.2)
```

```
##      WAIC pWAIC dWAIC weight    SE dSE
## m1.2.2  94.0   4.8   0.0   0.91 12.75  NA
## m2     98.6   4.0   4.6   0.09 13.32  4.75
```

*The model with interaction terms is much better than the one without, which is weighted for 0.92. The model without interaction terms is weighted for 0.08.*

*However, the coefficient of interaction terms is negative, which means that if the pirate eagle is adult and big simultaneously, the success probability would decrease. In addition, the intercept also turned negative after considering interaction terms, meaning that now for small eagles and when the pirate one is not adult, the success rate has decreased compared with previous model.*

*#Plot the PI*

```
eagles.pred.2 <- ensemble( m1 , m1.2, data=d.pred )
```

```
## Constructing posterior predictions
```

```

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

## Aggregated binomial counts detected. Splitting to 0/1 outcome for WAIC cal
culation.
## Constructing posterior predictions

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

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

eagles.med.2 <- apply( eagles.pred.2$link , 2 , median )
eagles.PI.2 <- apply( eagles.pred.2$link , 2 , PI )

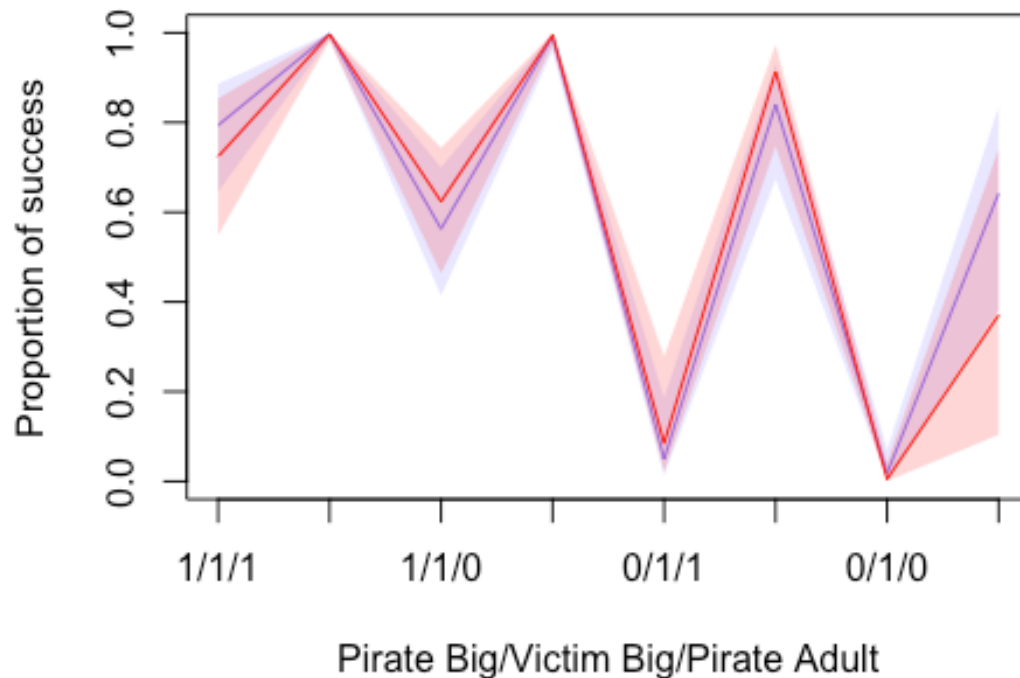
# plot predicted trend for high contact islands
plot( 0 , 0 , type="n" , xlab="Pirate Big/Victim Big/Pirate Adult" ,
      ylab="Proportion of success" , ylim=c(0,1) , xaxt="n" ,
      xlim=c(1,8) )
axis( 1 , at=1:8 , labels=c("1/1/1","1/0/1","1/1/0","1/0/0","0/1/1","0/0/1","
0/1/0","0/0/0") )

lines( 1:8 , eagles.med , col=rangi2 )
shade( eagles.PI , 1:8 , col=col.alpha(rangi2,0.2) )

lines( 1:8 , eagles.med.2 , col='red' )
shade( eagles.PI.2 , 1:8 , col=col.alpha('red',0.2) )

```





*From the shady area comparison, we can also notice that when using weighted averaged result, the red shady area is much bigger when 0/0/0, which is when both pirate and victim are small, and pirate is not adult. This means that the success might be much lower than previous model since we consider more uncertainty.*

## Question 2

The data contained in `data(salamanders)` are counts of salamanders (*Plethodon elongates*) from 47 different 49-m<sup>2</sup> plots in northern California. The column `SALAMAN` is the count in each plot, and the columns `PCTCOVER` and `FORESTAGE` are percent of ground cover and age of trees in the plot, respectively. You will model `SALAMAN` as Poisson variable.

(a) Model the relationship between density and percent cover, using a log-link (same as the example in the textbook and lecture). Use weakly informative priors of your choosing. Check the quadratic approximation again, by comparing `map` to `map2stan`.

```
data(salamanders)
d2=salamanders
```

```

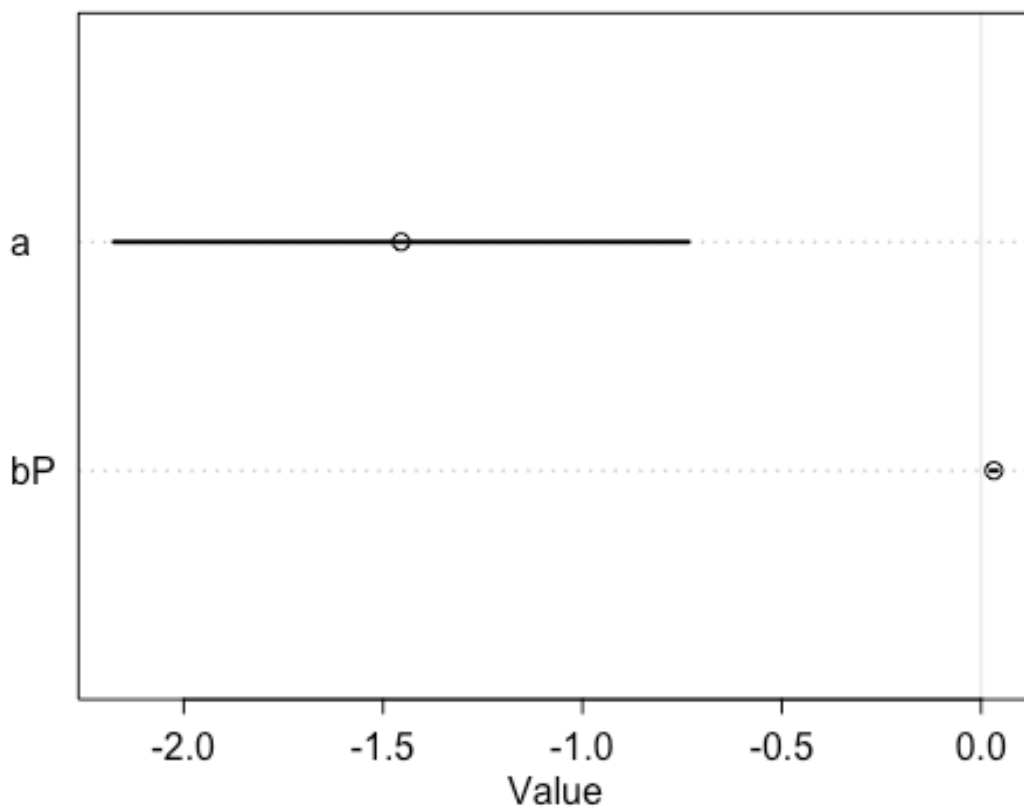
m3 <- map(
  alist(
    SALAMAN ~ dpois( lambda ),
    log(lambda) <- a + bP*PCTCOVER,
    a ~ dnorm(0,100),
    bP ~ dnorm(0,1)
  ),
  data=d2 )

precis(m3,corr=TRUE)

##      Mean StdDev  5.5% 94.5%      a      bP
## a  -1.45   0.45 -2.18 -0.73   1.00 -0.98
## bP  0.03   0.01  0.02  0.04 -0.98  1.00

plot(precis(m3))

```



```

m3.2 = map2stan(
  m3,data=d2 , iter=3000 , warmup=1000 , chains=4 )

## Constructing posterior predictions

```

```
## [ 800 / 8000 ]
[ 1600 / 8000 ]
[ 2400 / 8000 ]
[ 3200 / 8000 ]
[ 4000 / 8000 ]
[ 4800 / 8000 ]
[ 5600 / 8000 ]
[ 6400 / 8000 ]
[ 7200 / 8000 ]
[ 8000 / 8000 ]

precis(m3.2)

##      Mean StdDev lower 0.89 upper 0.89 n_eff Rhat
## a   -1.56   0.46    -2.26    -0.80  1138    1
## bP   0.03   0.01     0.02     0.04  1161    1

coefstab(m3,m3.2)

##      m3      m3.2
## a    -1.45    -1.56
## bP     0.03     0.03
## nobs     47     47
```

Then plot the expected counts and their 89% interval against percent cover. In which ways does the model do a good job? In which ways does it do a bad job?

## Plot the 89%

```
plot( dPCTCOVER, dSALAMAN , col=rangi2 , pch=pch , xlab="Ground Cover%" , ylab="# of Salamanders")
```

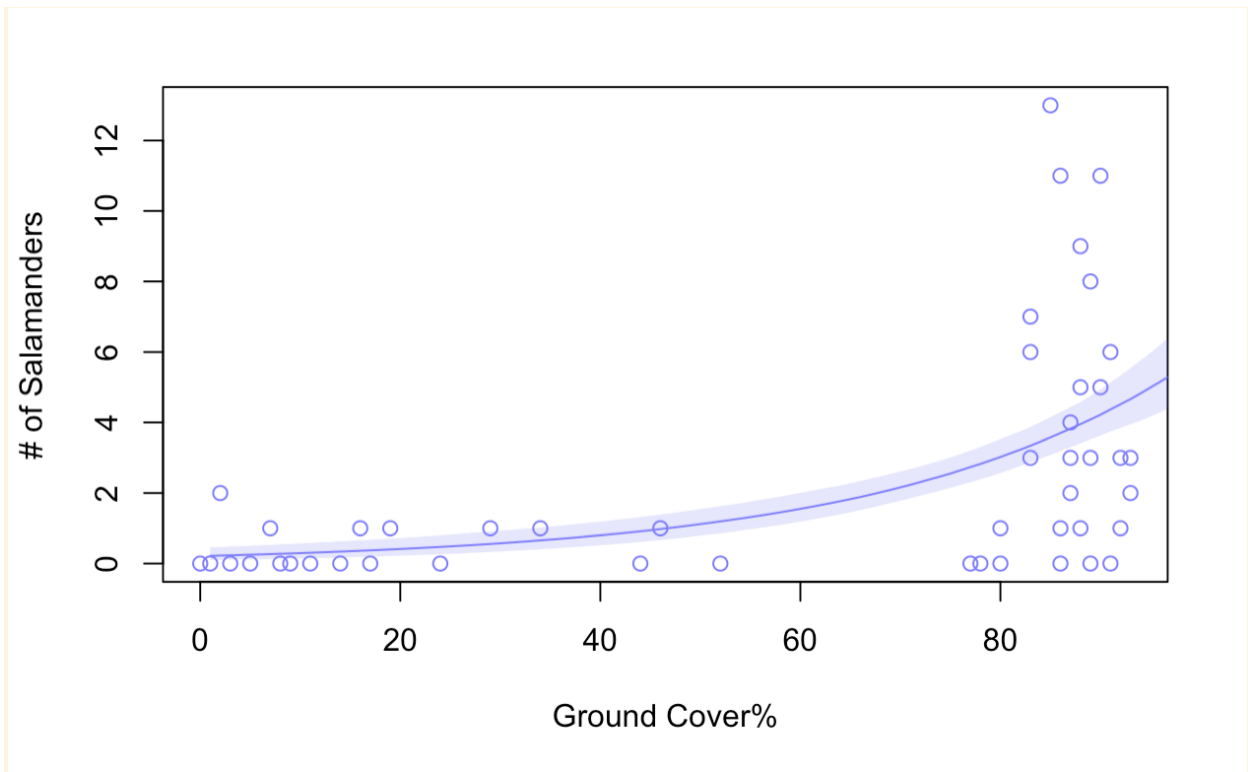
```
percov.seq <- seq( from=1 , to=100 , by=1 )
```

## compute 1~100% of ground cover percentage

```
d.pred <- data.frame( PCTCOVER = percov.seq ) sal.pred <- ensemble( m3.2 , data=d.pred )
sal.med <- apply( sal.pred$link, 2, median) sal.PI <- -apply( sal.pred$link , 2 , PI )
```

## plot predicted trend for high contact islands

```
lines( percov.seq , sal.med , col=rangi2 ) shade( sal.PI , percov.seq ,
col=col.alpha(rangi2,0.2) )
```



*The model does well when predicting the ground cover % around 0%~50%. However, when ground cover % is around 80~100, the predicted value is poor.*

**Can you improve the model by using other predictors, FORESTAGE? Try any models you think useful. Can you explain why the FORESTAGE helps or does not help with prediction?**

```
library(rethinking) m4 <- map2stan( alist( SALAMAN ~ dpois( lambda ), log(lambda) <- a + bPPCTCOVER + bFFORESTAGE, a ~ dnorm(0,100), bP ~ dnorm(0,1), bF ~ dnorm(0,1) ), data=d2, iter=5000 , warmup=1000 , chains=2 ) ) precis(m4)
```

	Mean	StdDev	lower 0.89	upper 0.89	n_eff	Rhat
a	-1.57	0.48	-2.28	-0.80	1530	1
bP	0.03	0.01	0.02	0.04	1505	1
bF	0.00	0.00	0.00	0.00	4313	1

```
compare(m3.2,m4,m4.2) coeftab(m3.2,m4,m4.2)
```

```
m4.3 <- map2stan( alist( SALAMAN ~ dpois( lambda ), log(lambda) <- a + bPPCTCOVER + bPPPCTCOVERPCTCOVER+ bFFORESTAGE, a ~ dnorm(0,10), bP ~ dnorm(0,1), bPP ~ dnorm(0,1), bF ~ dnorm(0,1) ), data=d2, iter=5000 , warmup=1000 , chains=2 ) ) precis(m4.3)
```

```
compare(m3.2,m4,m4.2,m4.3) coeftab(m3.2,m4,m4.2,m4.3)
```

```
m4.4 <- map2stan( alist( SALAMAN ~ dpois( lambda ), log(lambda) <- a[SITE] +
bPPCTCOVER + bPPPCTCOVER*PCTCOVER, a[SITE] ~ dnorm(0,10), bP ~ dnorm(0,1), bPP
~ dnorm(0,1) ), data=d2, iter=5000 , warmup=1000 , chains=2 ) precis(m4.4)
compare(m3.2,m4,m4.2,m4.3,m4.4)
```

	WAIC	pWAIC	dWAIC	weight	SE	dSE
m4.4	143.9	17.9	0.0	1	15.83	NA
m3.2	213.4	4.7	69.5	0	26.45	24.35
m4.2	216.0	6.6	72.1	0	26.98	24.66
m4	217.7	7.6	73.8	0	27.32	24.99

m4.4 would be the best model

```
#####Plot the new prediction with squared term of Ground cover %
```

```
plot( d$PCTCOVER , d$SALAMAN , col=rangi2 , pch=pch ,
      xlab="Ground Cover%" , ylab="# of Salamanders" )
```

```
percov.seq <- seq( from=1 , to=100 , length.out = 47 )
```

```
# compute 1~100% of ground cover percentage
```

```
d.pred <- data.frame(
  PCTCOVER = percov.seq,
  SITE = 1:47
)
```

```
sal.pred <- ensemble( m4.2 , data=d.pred )
```

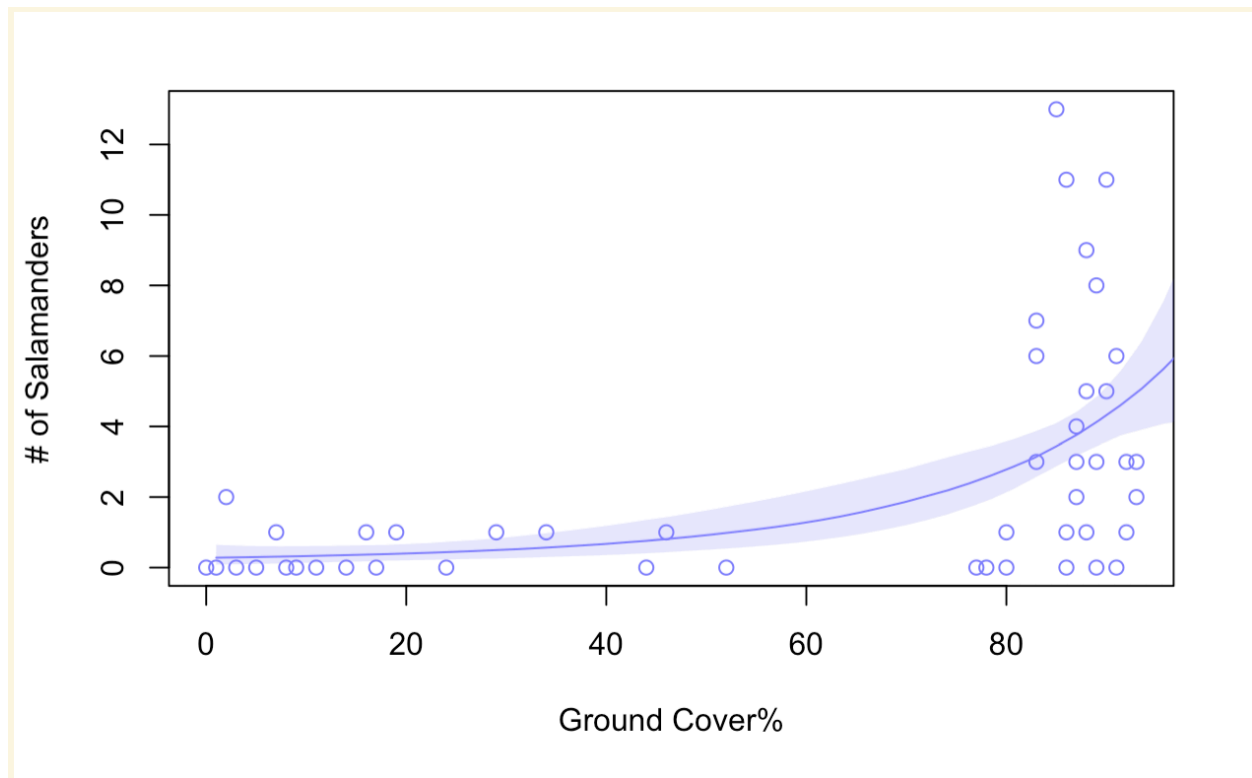
```
sal.med <- apply( sal.pred$link , 2 , median )
```

```
sal.PI <- apply( sal.pred$link , 2 , PI )
```

```
# plot predicted trend for high contact islands
```

```
lines( percov.seq , sal.med , col=rangi2 )
```

```
shade( sal.PI , percov.seq , col=col.alpha(rangi2,0.2) )
```



*The prediction still can't fit well*

**Allow intercept differ in every individual**

```
plot( dPCTCOVER, dSALAMAN , col=rangi2 , pch=pch , xlab="Ground Cover%" , ylab="# of Salamanders" )
```

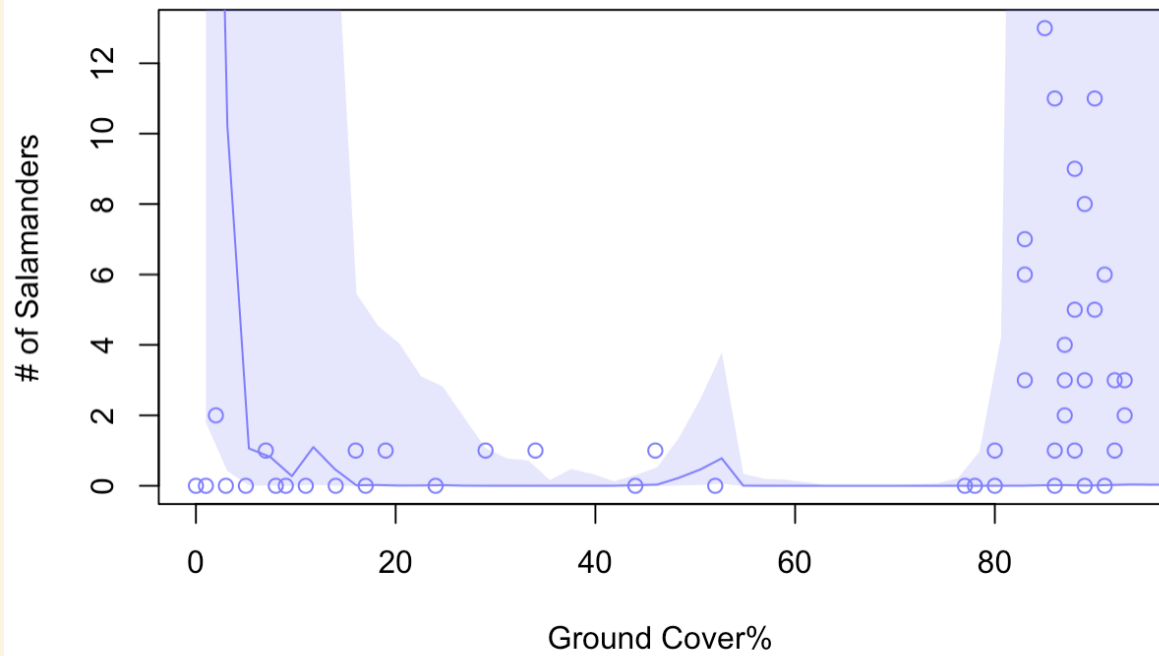
```
percov.seq <- seq( from=1 , to=100 , length.out = 47 )
```

**compute 1~100% of ground cover percentage**

```
d.pred <- data.frame( PCTCOVER = percov.seq, SITE = 1:47 ) sal.pred <- ensemble( m4.4 ,
data=d.pred ) sal.med <- apply( sal.predlink, 2, median) sal.PI <- apply( sal.predlink, 2 ,
PI )
```

**plot predicted trend of allowing different intercept individually**

```
lines( percov.seq , sal.med , col=rangi2 ) shade( sal.PI , percov.seq ,
col=col.alpha(rangi2,0.2) ) ``
```



*The prediction fitted well around 80%~100%, but fitted poorly around 0%~40%*

*Perhaps it would be better to split group into 0%~60% and 80%~100% to make better prediction.*