JAGS Model

Tom Park

2020-02-08

Basic Model: Ambulance + Overdose

Ambulance Call-outs Model

n_A: sample size
x_A: the total number who confirmed they did call an ambulance
p_A: probability of a person call an ambulance

$$x_A \sim Bin(n_A, p_A)$$

We assume $n_A = 1000, p_A = 0.8$.
Suppose the prior of p_A is noninformative.

$$p(p_A) \sim Beta(1, 1)$$

Overdose Model

Now we plug in this values into the overdose model and obtain possible O_t values **assuming we have U_t values.**

Also, we have priors.

$$z_t \sim N(\mu, \sigma^2)$$
$$\lambda_t^{OD} = \exp(z_t)$$
$$O_t \sim Poi(\lambda_t^{OD} N)$$
$$U_t \sim Bin(O_t, p_A)$$

For simplicity we set N =10000 for now. We need to generate reasonable U_t values first. Note that U_t comes from μ, σ following all the way through the overdose model.
$\mu = \log 0.05, \sigma = 1, N = 10000$.
We suppose survey data exists: (n_A, x_A) known.

We set for our prior parameters:

$$\mu \sim U(-10, 0)$$
$$\sigma \sim U(0, 5)$$

```
# install packages
if (!require(rjags)) install.packages("rjags", dependencies = TRUE)
if (!require(coda)) install.packages("coda", dependencies = TRUE)
if (!require(tidyverse)) install.packages("tidyverse", dependencies = TRUE)
if (!require(tinytex)) install.packages("tinytex", dependencies = TRUE)

library('rjags')
library('coda')
library('tidyverse')
library('tinytex')
```

The data is the same data from pymc3 with Python.
Todo: build a pipeline to connect the python (pymc3) and R (JAGS)

```r
df <- read.csv('./basic_data.csv')
df$X <- NULL
head(df)
```

```
##    o_t  u_t x_a
## 1 2475 1969 799
## 2  262  217 798
## 3  318  253 795
## 4  149  119 816
## 5 1151  934 805
## 6   39   34 794
```

Now we set the model which defines the relations of overdose model and ambulance call model.

The model defined as follows.

```r
cat("model{
## define the priors
p_a ~ dbeta(alpha, beta)
mu ~ dunif(mu_a, mu_b)
sigma ~ dunif(sigma_a, sigma_b)

## the latent variables
z ~ dnorm(mu, 1/(sigma^2))
lambda <- exp(z)

for (i in 1:n) {
  ## ambulance model
  x_a[i] ~ dbin(p_a, n_a) # each survey result for month
}
for (i in 1:n) {
  ## overdose model
  o_t[i] ~ dpois(lambda*N) # total overdoses per month
}
for (i in 1:n) {
  u_t[i] ~ dbin(p_a, o_t[i]) # ambulanced overdoses per month
}

}", file='basic_model.txt')
```

Pre-set variables.

```r
n <-  length(df$o_t) # we have 12 samples
n_a <- 1000
N <- 10000
u_t <- df$u_t
x_a <- df$x_a
```

Define the list providing the values of the variables and the parameters for the priors of the model.

```r
dat <- list(
            # priors for ambulance model
            'alpha' = 1,
            'beta' =  1,
```

```
             # priors for overdose model
             'mu_a'=(-10),
             'mu_b'=0,
             'sigma_a'=0,
             'sigma_b'=5,

             # likelihood
             'u_t'=u_t,
             'x_a'=x_a,
             'N'=N, # the population
             'n'= n, # total months
             'n_a'=n_a

             )
```

Note: for the list object usually named 'data' or 'dat' in JAGS context, do not use arrow but use equal sign to define elements of the list.

```
interations = 1000
burnin= floor(interations/2)
chains=2
# inits = list()
simple.model <- jags.model(file='basic_model.txt',
                           data=dat,
                           n.chains = chains)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 24
##    Unobserved stochastic nodes: 16
##    Total graph size: 55
##
## Initializing model
```

```
simple.model
```

```
## JAGS model:
##
## model{
## ## define the priors
## p_a ~ dbeta(alpha, beta)
## mu ~ dunif(mu_a, mu_b)
## sigma ~ dunif(sigma_a, sigma_b)
##
## ## the latent variables
## z ~ dnorm(mu, 1/(sigma^2))
## lambda <- exp(z)
##
## for (i in 1:n) {
##   ## ambulance model
##   x_a[i] ~ dbin(p_a, n_a) # each survey result for month
## }
## for (i in 1:n) {
##   ## overdose model
```

```
##   o_t[i] ~ dpois(lambda*N) # total overdoses per month
## }
## for (i in 1:n) {
##   u_t[i] ~ dbin(p_a, o_t[i]) # ambulanced overdoses per month
## }
##
## }
## Fully observed variables:
##  N alpha beta mu_a mu_b n n_a sigma_a sigma_b u_t x_a
```

O_t

```
params= c('o_t')
samples <- coda.samples(simple.model, params, n.iter = 1000)
```

Q1: is init necessary? and it's a characteristic of Gibbs sampling?

```
summary(window(samples), start=burnin)
```

```
##
## Iterations = 1001:2000
## Thinning interval = 1
## Number of chains = 2
## Sample size per chain = 1000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean    SD Naive SE Time-series SE
## o_t[1]  2156.8 14.46   0.3232         0.3800
## o_t[2]   403.9 14.24   0.3185         0.3751
## o_t[3]   440.6 14.39   0.3218         0.3792
## o_t[4]   306.0 14.14   0.3162         0.4288
## o_t[5]  1121.1 14.75   0.3298         0.3921
## o_t[6]   221.8 14.45   0.3232         0.3769
## o_t[7]  2579.2 14.41   0.3222         0.3525
## o_t[8]   383.5 14.35   0.3209         0.3939
## o_t[9]   756.2 14.63   0.3271         0.3968
## o_t[10]  495.3 14.68   0.3283         0.4205
## o_t[11] 1842.2 14.68   0.3283         0.3985
## o_t[12]  241.8 14.84   0.3319         0.4170
##
## 2. Quantiles for each variable:
##
##         2.5%  25%  50%  75% 97.5%
## o_t[1]  2129 2147 2157 2167  2186
## o_t[2]   376  394  404  413   433
## o_t[3]   413  431  440  450   469
## o_t[4]   280  296  306  315   336
## o_t[5]  1093 1111 1121 1131  1152
## o_t[6]   194  212  221  232   250
## o_t[7]  2551 2569 2579 2589  2608
## o_t[8]   356  374  383  393   412
```

```
## o_t[9]    728  746  756  766   785
## o_t[10]   466  486  495  505   524
## o_t[11] 1815 1832 1842 1852  1871
## o_t[12]   213  232  241  252   271
```

Boxplots of O__t

q2: I see two elements from the samples list. Which one I should use it or should I use both?

```
temp = as.matrix(samples)
colnames(temp) <- seq(1,12)
head(temp)
```

```
##          1   2   3   4    5   6    7   8   9  10   11  12
## [1,] 2160 375 456 299 1101 222 2613 381 744 523 1862 244
## [2,] 2150 384 447 309 1123 203 2579 386 750 493 1839 265
## [3,] 2151 421 462 326 1100 216 2563 400 762 462 1819 225
## [4,] 2147 398 457 290 1126 202 2581 379 741 496 1844 234
## [5,] 2156 410 426 323 1113 212 2575 385 741 491 1839 223
## [6,] 2154 386 452 297 1146 218 2582 375 768 486 1838 229
```

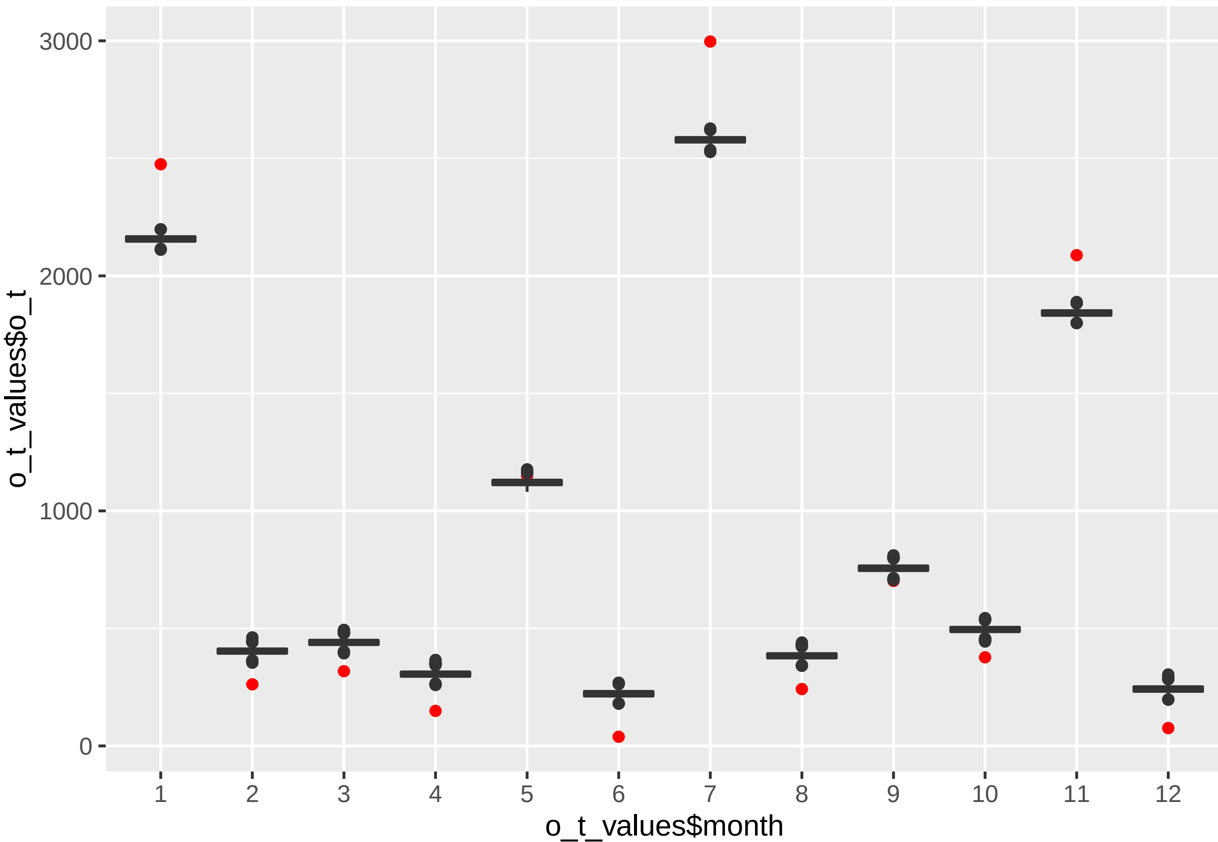
```
df_o_t <- as.data.frame(temp)
head(df_o_t)
```

```
##      1   2   3   4    5   6    7   8   9  10   11  12
## 1 2160 375 456 299 1101 222 2613 381 744 523 1862 244
## 2 2150 384 447 309 1123 203 2579 386 750 493 1839 265
## 3 2151 421 462 326 1100 216 2563 400 762 462 1819 225
## 4 2147 398 457 290 1126 202 2581 379 741 496 1844 234
## 5 2156 410 426 323 1113 212 2575 385 741 491 1839 223
## 6 2154 386 452 297 1146 218 2582 375 768 486 1838 229
```

```
df_o_t <- gather(df_o_t, key = 'month',value = 'o_t')
df_o_t$month <- factor(df_o_t$month,levels = seq(1,12))
str(df_o_t)
```

```
## 'data.frame':    24000 obs. of  2 variables:
##  $ month: Factor w/ 12 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ o_t  : num  2160 2150 2151 2147 2156 ...
```

```
o_t_values=data.frame('month'=seq(1,12),'o_t'=df$o_t)
o_t_values$month <- factor(o_t_values$month,levels = seq(1,12))

ggplot()+geom_point(aes(x=o_t_values$month, y=o_t_values$o_t),color='red')+geom_boxplot(aes(x=month,y=o_
```



Predictive Posterior Checks

```
params= c('u_t','x_a')
ppc <- coda.samples(simple.model, params, n.iter = 1000)
```

```
summary(window(ppc),start=burnin)
```

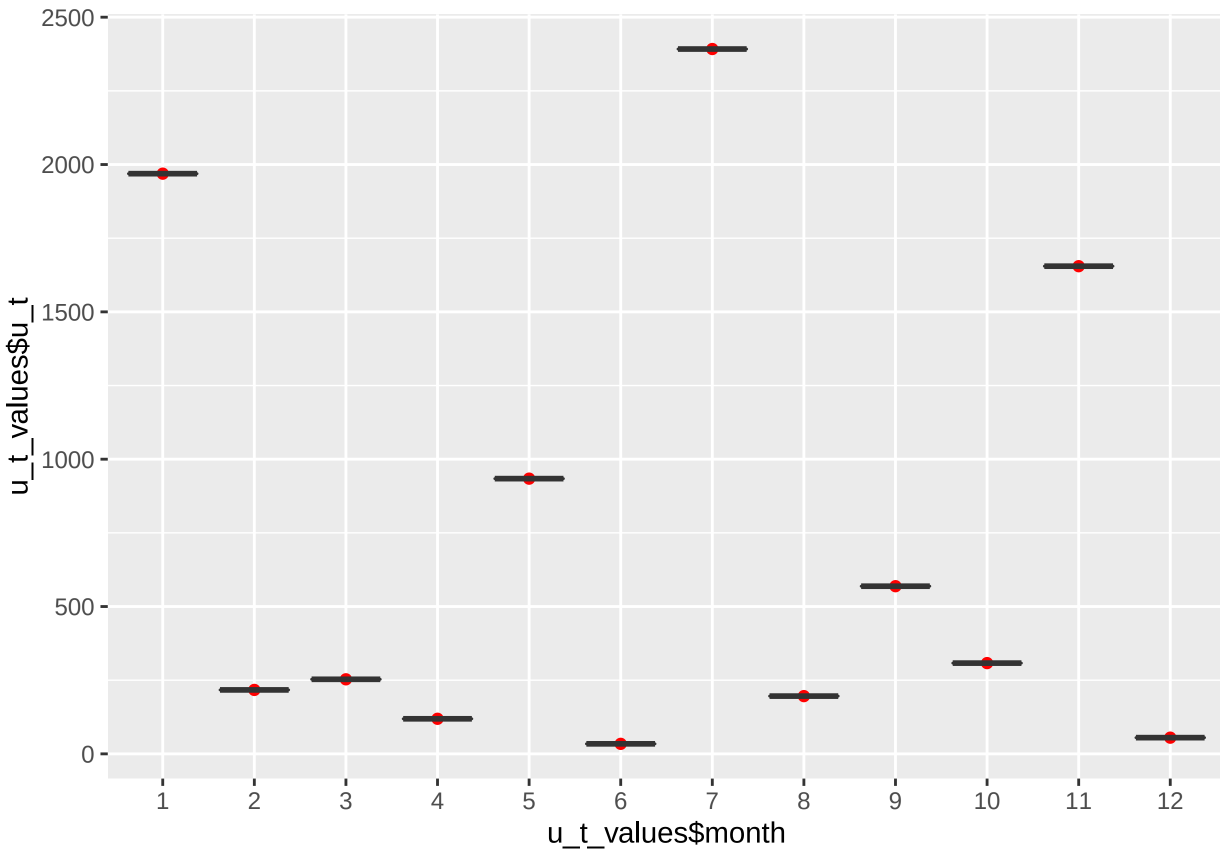
```
##
## Iterations = 2001:3000
## Thinning interval = 1
## Number of chains = 2
## Sample size per chain = 1000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##          Mean SD Naive SE Time-series SE
## u_t[1]   1969  0        0              0
## u_t[2]    217  0        0              0
## u_t[3]    253  0        0              0
## u_t[4]    119  0        0              0
## u_t[5]    934  0        0              0
## u_t[6]     34  0        0              0
## u_t[7]   2392  0        0              0
## u_t[8]    196  0        0              0
## u_t[9]    569  0        0              0
## u_t[10]   308  0        0              0
## u_t[11]  1655  0        0              0
```

```
## u_t[12]   55  0        0              0
## x_a[1]   799  0        0              0
## x_a[2]   798  0        0              0
## x_a[3]   795  0        0              0
## x_a[4]   816  0        0              0
## x_a[5]   805  0        0              0
## x_a[6]   794  0        0              0
## x_a[7]   793  0        0              0
## x_a[8]   780  0        0              0
## x_a[9]   773  0        0              0
## x_a[10]  779  0        0              0
## x_a[11]  788  0        0              0
## x_a[12]  813  0        0              0
##
## 2. Quantiles for each variable:
##
##          2.5%  25%  50%  75% 97.5%
## u_t[1]   1969 1969 1969 1969  1969
## u_t[2]    217  217  217  217   217
## u_t[3]    253  253  253  253   253
## u_t[4]    119  119  119  119   119
## u_t[5]    934  934  934  934   934
## u_t[6]     34   34   34   34    34
## u_t[7]   2392 2392 2392 2392  2392
## u_t[8]    196  196  196  196   196
## u_t[9]    569  569  569  569   569
## u_t[10]   308  308  308  308   308
## u_t[11]  1655 1655 1655 1655  1655
## u_t[12]    55   55   55   55    55
## x_a[1]    799  799  799  799   799
## x_a[2]    798  798  798  798   798
## x_a[3]    795  795  795  795   795
## x_a[4]    816  816  816  816   816
## x_a[5]    805  805  805  805   805
## x_a[6]    794  794  794  794   794
## x_a[7]    793  793  793  793   793
## x_a[8]    780  780  780  780   780
## x_a[9]    773  773  773  773   773
## x_a[10]   779  779  779  779   779
## x_a[11]   788  788  788  788   788
## x_a[12]   813  813  813  813   813
```

U_t: Predictive Posterior Checks

```r
temp = as.matrix(ppc)

df_u_t=temp[,1:12]
df_x_a=temp[,13:24]


test <- as.matrix(df_u_t)
colnames(test) <- seq(1,12)
mtx_u_t <- as.data.frame(test)
```

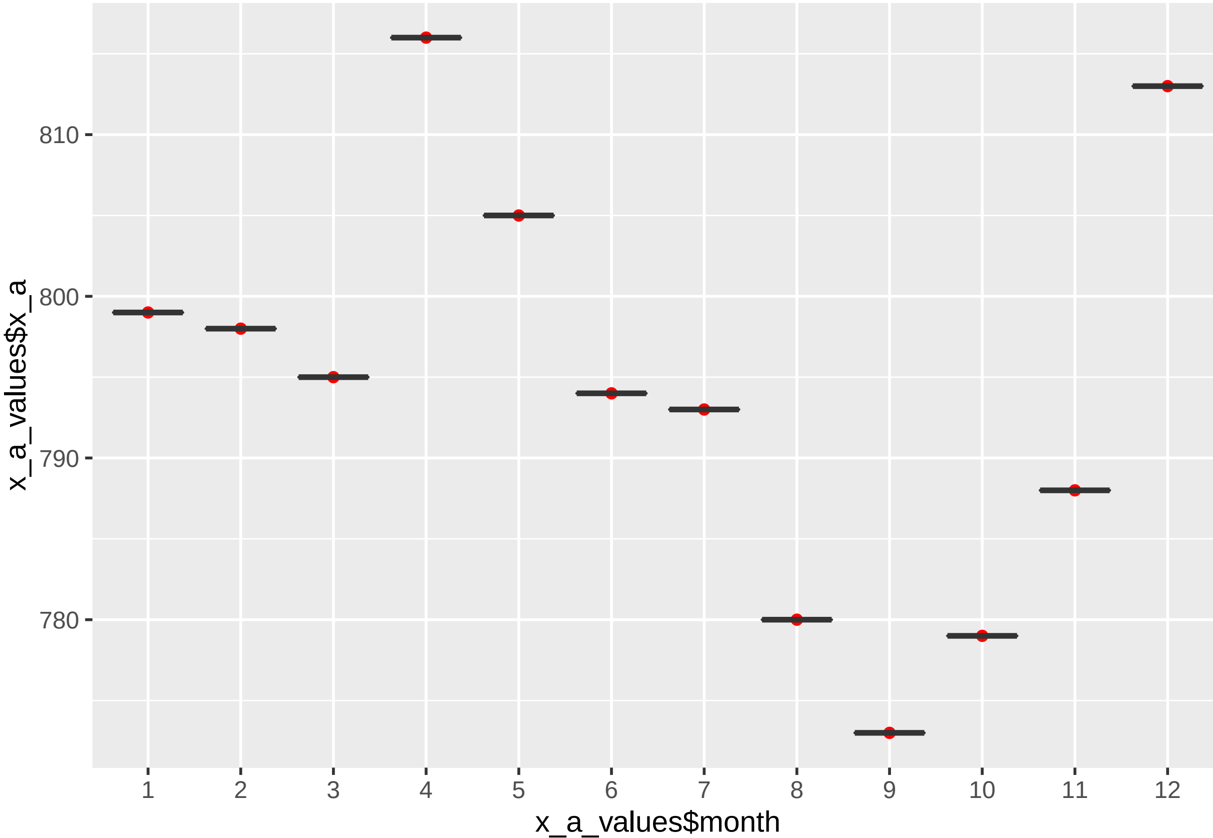
```
df_u_t <- gather(mtx_u_t, key = 'month',value = 'u_t')

df_u_t$month <- factor(df_u_t$month,levels = seq(1,12))

u_t_values=data.frame('month'=seq(1,12),'u_t'=df$u_t)
u_t_values$month <- factor(u_t_values$month,levels = seq(1,12))

ggplot()+geom_point(aes(x=u_t_values$month, y=u_t_values$u_t),color='red')+geom_boxplot(aes(x=month,y=u
```



x_A: Predictive Posterior Checks

```
temp = as.matrix(ppc)
df_x_a=temp[,13:24]


test <- as.matrix(df_x_a)
colnames(test) <- seq(1,12)

mtx_x_a <- as.data.frame(test)


df_x_a <- gather(mtx_x_a, key = 'month',value = 'x_a')


df_x_a$month <- factor(df_x_a$month,levels = seq(1,12))
```

```
x_a_values=data.frame('month'=seq(1,12),'x_a'=df$x_a)
x_a_values$month <- factor(x_a_values$month,levels = seq(1,12))

ggplot()+geom_point(aes(x=x_a_values$month, y=x_a_values$x_a),color='red')+geom_boxplot(aes(x=month,y=x_
```



Reference

first tutorial
second tutorial
JAGS manual
error handling guide