

Appendix

Chloe You
xinyuan.you@stat.ubc.ca

April 2022

Simulation: Normal, Student-t, and Cauchy Prior Comparisons

```
library(MCMCvis)
library(rjags)
set.seed(1)
```

Generate Data

Let x_i be the indicator variable of being in the treatment group or not. $p_{control}$ and $p_{trtment}$ is the true probability of the subject being in each of the group. We simulate data via the following process:

$$\begin{aligned}x_i &\sim \text{Bernoulli}(0.5) \\ y_i &\sim \text{Binomial}(n, p_i) \\ p_i &= \begin{cases} p_{control} & x_i = 0 \\ p_{trtment} = p_{control}RR_{true} & x_i = 1 \end{cases}\end{aligned}$$

We set sample size $n = 100$, $p_{control} = 0.3$, and $RR_{true} = 0.7$. We set $\beta_0 = -2$ for easier computation.

```
sim = list()

init = 1
for(r in c(0.7)){
  for (n in c(100)){
    sim[[init]] = data.frame(x=numeric(n),p=numeric(n),y=numeric(n))
    for(i in c(1:n)){
      sim[[init]]$x[i] = rbinom(1,1,prob=0.5)
      sim[[init]]$p[i] = (1-sim[[init]]$x[i])*0.3 + sim[[init]]$x[i]*0.3*r
      sim[[init]]$y[i] = rbinom(1, 1, prob=sim[[init]]$p[i])
    }
    init = init + 1
  }
}
```

Prior Specification and Statistical Model

Prior 1: $\beta_1 \sim \text{Normal}(0, 0.35^2)$

```
genmod.string <- "model{

  ### prior distribution
  b1 ~ dnorm(0, 8.163)

  ### target parameter
  RR <- exp(b1)

  ### statistical model
  for (i in 1:n) {
    y[i] ~ dbern(p.bound[i])
    p.bound[i] <- max(0, min(1, p[i]))
    log(p[i]) <- -2 + b1*x[i]
  }
}"

### generative model, data go in

mod <- jags.model(textConnection(genmod.string),
                  data=list(x= sim[[1]]$x, y= sim[[1]]$y,n= nrow(sim[[1]])),
                  n.chains=3)

## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 100
##   Unobserved stochastic nodes: 1
##   Total graph size: 218
##
## Initializing model

# n.chains = the number of parallel chains for the model

### MC output comes out

opt.JAGS <- coda.samples(mod,
                          variable.names=c("RR"),
                          n.iter=10000)

MCMCsummary(opt.JAGS)

##           mean          sd      2.5%      50%      97.5% Rhat n.eff
## RR 0.9850295 0.2409071 0.5834431 0.9630749 1.517265    1 18797
```

Prior 2: $\beta_1 \sim \text{Cauchy}(0, 2.5^2)$

```
genmod.string <- "model{  
  
  ### prior distribution  
  b1 ~ dt(0, 0.16, 1)  
  
  ### target parameter  
  RR <- exp(b1)  
  
  ### statistical model  
  for (i in 1:n) {  
    y[i] ~ dbern(p.bound[i])  
    p.bound[i] <- max(0, min(1, p[i]))  
    log(p[i]) <- -2 + b1*x[i]  
  }  
}"  
  
### generative model, data go in  
  
mod <- jags.model(textConnection(genmod.string),  
                  data=list(x= sim[[1]]$x, y= sim[[1]]$y,n= nrow(sim[[1]])),  
                  n.chains=3)
```

```
## Compiling model graph  
##   Resolving undeclared variables  
##   Allocating nodes  
## Graph information:  
##   Observed stochastic nodes: 100  
##   Unobserved stochastic nodes: 1  
##   Total graph size: 218  
##  
## Initializing model
```

```
# n.chains = the number of parallel chains for the model
```

```
### MC output comes out
```

```
opt.JAGS <- coda.samples(mod,  
                          variable.names=c("RR"),  
                          n.iter=10000)
```

```
MCMCsummary(opt.JAGS)
```

```
##           mean          sd      2.5%      50%      97.5% Rhat n.eff  
## RR 0.9461957 0.3243419 0.4158416 0.9113017 1.667648    1 18951
```

Prior 3: $\beta_1 \sim t_7(0, 0.35^2)$

```

genmod.string <- "model{

  ### prior distribution
  b1 ~ dt(0, 8.16, 7)

  ### target parameter
  RR <- exp(b1)

  ### statistical model
  for (i in 1:n) {
    y[i] ~ dbern(p.bound[i])
    p.bound[i] <- max(0, min(1, p[i]))
    log(p[i]) <- -2 + b1*x[i]
  }
}"

### generative model, data go in

mod <- jags.model(textConnection(genmod.string),
                  data=list(x= sim[[1]]$x, y= sim[[1]]$y,n= nrow(sim[[1]])),
                  n.chains=3)

```

```

## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 100
##   Unobserved stochastic nodes: 1
##   Total graph size: 219
##
## Initializing model

```

n.chains = the number of parallel chains for the model

```

### MC output comes out

opt.JAGS <- coda.samples(mod,
                          variable.names=c("RR"),
                          n.iter=10000)

MCMCsummary(opt.JAGS)

```

```

##           mean      sd      2.5%      50%      97.5% Rhat n.eff
## RR 0.9849256 0.244594 0.5699561 0.9640458 1.517404    1 18596

```

Non-Bayesian Estimates

```

nonbay= glm(formula = sim[[1]]$y ~ sim[[1]]$x, family = binomial(link = log))
summary(nonbay)

```

```
##
## Call:
## glm(formula = sim[[1]]$y ~ sim[[1]]$x, family = binomial(link = log))
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -0.9606  -0.9606  -0.5270  -0.5270   2.0214
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.9954      0.1926  -5.169 2.35e-07 ***
## sim[[1]]$x   -1.0476      0.4018  -2.608 0.00912 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 110.22  on 99  degrees of freedom
## Residual deviance: 102.26  on 98  degrees of freedom
## AIC: 106.26
##
## Number of Fisher Scoring iterations: 6
```

```
exp(confint(nonbay))
```

```
## Waiting for profiling to be done...
```

```
##              2.5 %    97.5 %
## (Intercept) 0.2399186 0.5133621
## sim[[1]]$x  0.1464106 0.7337500
```