# Markov Chain Monte Carlo
# Introduction to Statistics Using R (Psychology 9041B)

Paul Gribble

Winter, 2015

## Two Coins

Assume someone gives you two coins and you flip each 7 times, and get the following data:

| Coin 1 | Coin 2 |
|---|---|
| H | H |
| H | H |
| H | T |
| H | T |
| H | T |
| T | T |
| T | T |

Are the two coins both fair? In other words if you assume a bernoulli process, is the chance of getting a Heads on each coin flip equal to 0.50 for both coins? If not, what is the Pr(Heads)? Is there a difference between Pr(Heads) for Coin 1 and Coin 2?

### MCMC

Here we will use Markov Chain Monte Carlo to answer these questions.

We will model each coin flip as a Bernoulli process. As such we have a model of each coin, and the parameter we need to estimate for each coin is the probability of getting a Heads on each coin toss, Pr(Heads).

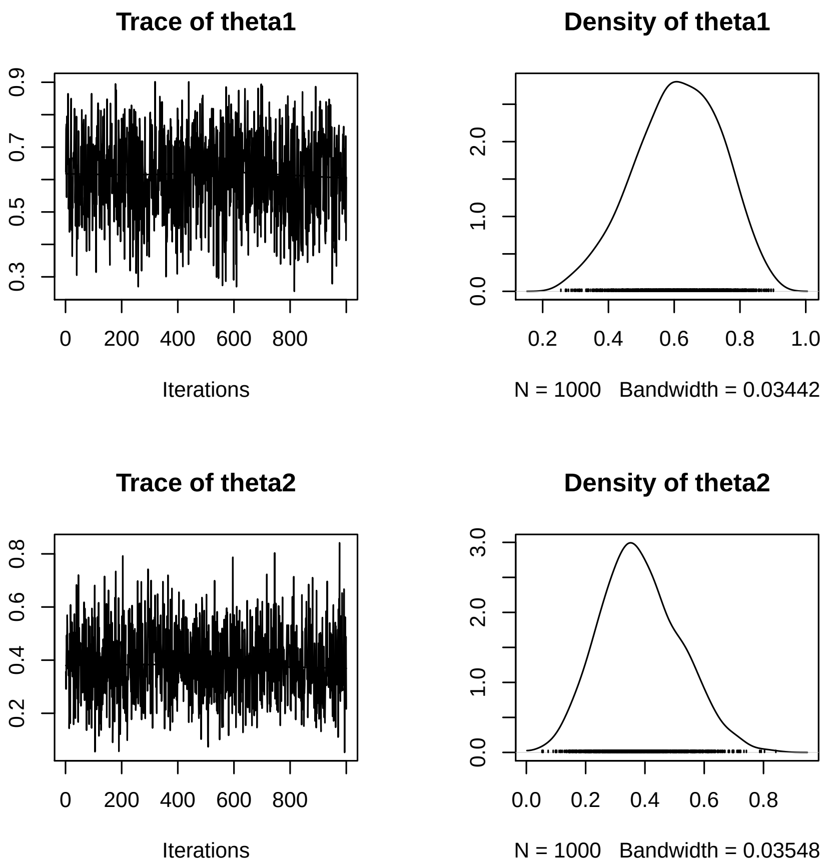
We will use the `rjags` package in `R` to model this.

Here is the file `http://www.gribblelab.org/stats/code/2coins.bug` which specifies the model for each coin:

```
model {
  # likelihood: each flip is bernoulli
  for (i in 1:N1) {y1[i] ~ dbern(theta1)}
  for (i in 1:N2) {y2[i] ~ dbern(theta2)}
  # prior: slight expectation of 0.5
  theta1 ~ dbeta(3,3)
  theta2 ~ dbeta(3,3)
}
```

Now the R code to implement this (`http://www.gribblelab.org/stats/code/2coins.R`):

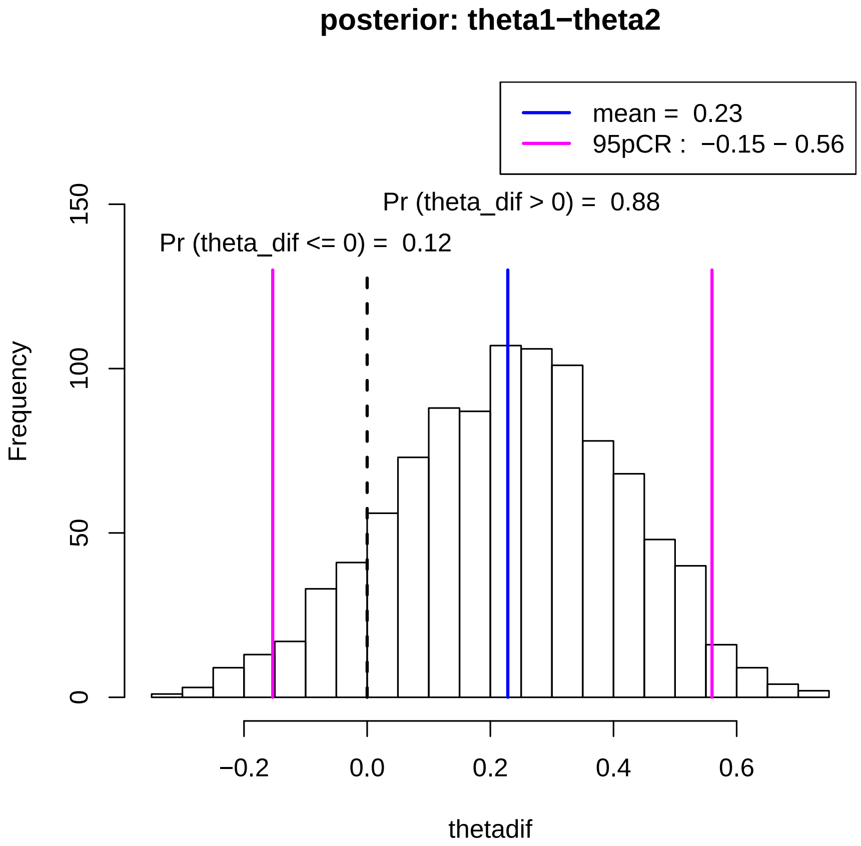
```
> # the data
> #
> y1 = c(1,1,1,1,1,0,0)
> y2 = c(1,1,0,0,0,0,0)
> N1 = length(y1)
> N2 = length(y2)
> # the model
> #
> library(rjags)
> jags = jags.model('../code/2coins.bug',
+ data = list('y1' = y1, 'y2' = y2, 'N1' = N1, 'N2' = N2),
+ n.chains = 1, n.adapt = 200)

Compiling model graph
   Resolving undeclared variables
   Allocating nodes
Graph information:
   Observed stochastic nodes: 14
   Unobserved stochastic nodes: 2
   Total graph size: 22

Initializing model

> # run mcmc sampling, generate 1000 samples
> # from the posterior for theta1 and theta2
> post = coda.samples(jags, c('theta1', 'theta2'), 1000)
> plot(post)
```



and then

```
> # plot credible differences in theta1 vs theta2
> #
> thetadif = post[[1]][,1] - post[[1]][,2]
> hist(thetadif,25,main="posterior: theta1-theta2",ylim=c(0,180))
> lines(c(0,0),c(0,130),col="black",lty=2,lwd=2)
> # mean difference
> #
> meandif = mean(thetadif)
> lines(c(meandif,meandif),c(0,130),col="blue",lty=1,lwd=2)
> # 95% credible range for difference between
> # theta1 & theta2
> #
> c95 = quantile(thetadif,c(.025, .975))
> lines(c(c95[1],c95[1]),c(0,130),col="magenta",lty=1,lwd=2)
> lines(c(c95[2],c95[2]),c(0,130),col="magenta",lty=1,lwd=2)
> legend(x="topright", col=c("blue","magenta"), lwd=c(2,2), lty=c(1,1),
+ legend=c(paste("mean = ",round(meandif,2)),
+ paste("95pCR : ", round(c95[1],2), "-", round(c95[2],2))))
```

```
> # prob thetadif > 0
> #
> prob_g_0 = length(which(thetadif>0)) / length(thetadif)
> text(0, 150, paste("Pr (theta_dif > 0) = ", round(prob_g_0,2)), pos=4)
> text(-.1, 145, paste("Pr (theta_dif <= 0) = ", round(1-prob_g_0,2)), pos=1)
```


posterior: theta1−theta2
mean = 0.23
95pCR : −0.15 − 0.56
Pr (theta_dif > 0) = 0.88
Pr (theta_dif <= 0) = 0.12
Frequency
thetadif