HW06.Rmd

Shashi

February 19, 2018

I have executed these exercises on my own and written the answers in my own words. Signed: Shashi Shankar

1A.

```
source("Jags-Ydich-Xnom1subj-MbernBetaModelComp.R")
```

```
##
## *******************************************************************
## Kruschke, J. K. (2015). Doing Bayesian Data Analysis, Second Edition:
## A Tutorial with R, JAGS, and Stan. Academic Press / Elsevier.
## *******************************************************************
```

```
## Loading required package: coda
```

```
## Linked to JAGS 4.2.0
```

```
## Loaded modules: basemod,bugs
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 0
##    Unobserved stochastic nodes: 18
##    Total graph size: 39
##
## Initializing model
##
## Burning in the MCMC chain...
## Sampling final MCMC chain...
```

Ran the script without the data i.e. y coomented in the model specification and got the prior as shown in the figure above.

1B.

```
source("Jags-Ydich-Xnom1subj-MbernBetaModelComp.R")
```

```
##
## *******************************************************************
## Kruschke, J. K. (2015). Doing Bayesian Data Analysis, Second Edition:
## A Tutorial with R, JAGS, and Stan. Academic Press / Elsevier.
## *******************************************************************
##
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
```

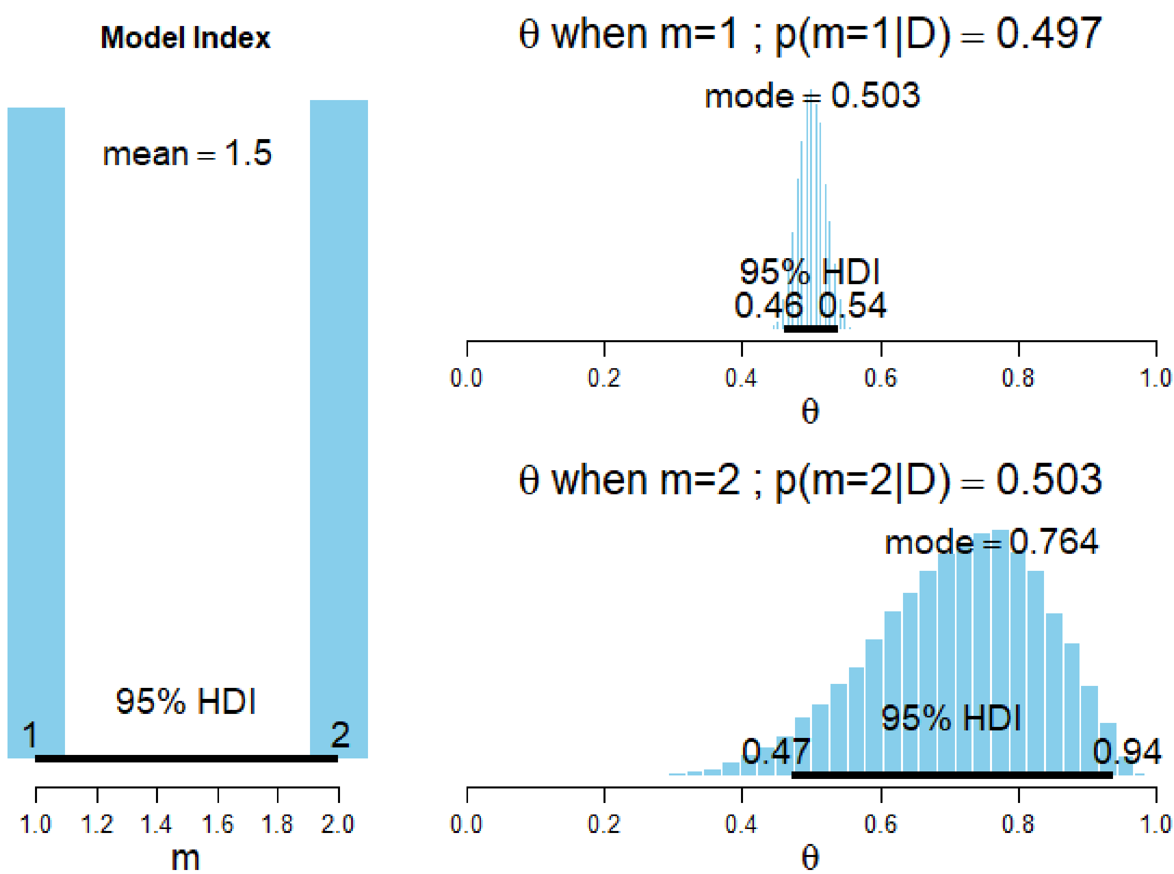
Model Index
mean = 1.5
95% HDI
1
2
1.0 1.2 1.4 1.6 1.8 2.0
m
θ when m=1 ; p(m=1|D) = 0.497
mode = 0.503
95% HDI
0.46 0.54
0.0 0.2 0.4 0.6 0.8 1.0
θ
θ when m=2 ; p(m=2|D) = 0.503
mode = 0.764
95% HDI
0.47
0.94
0.0 0.2 0.4 0.6 0.8 1.0
θ

Figure 1:

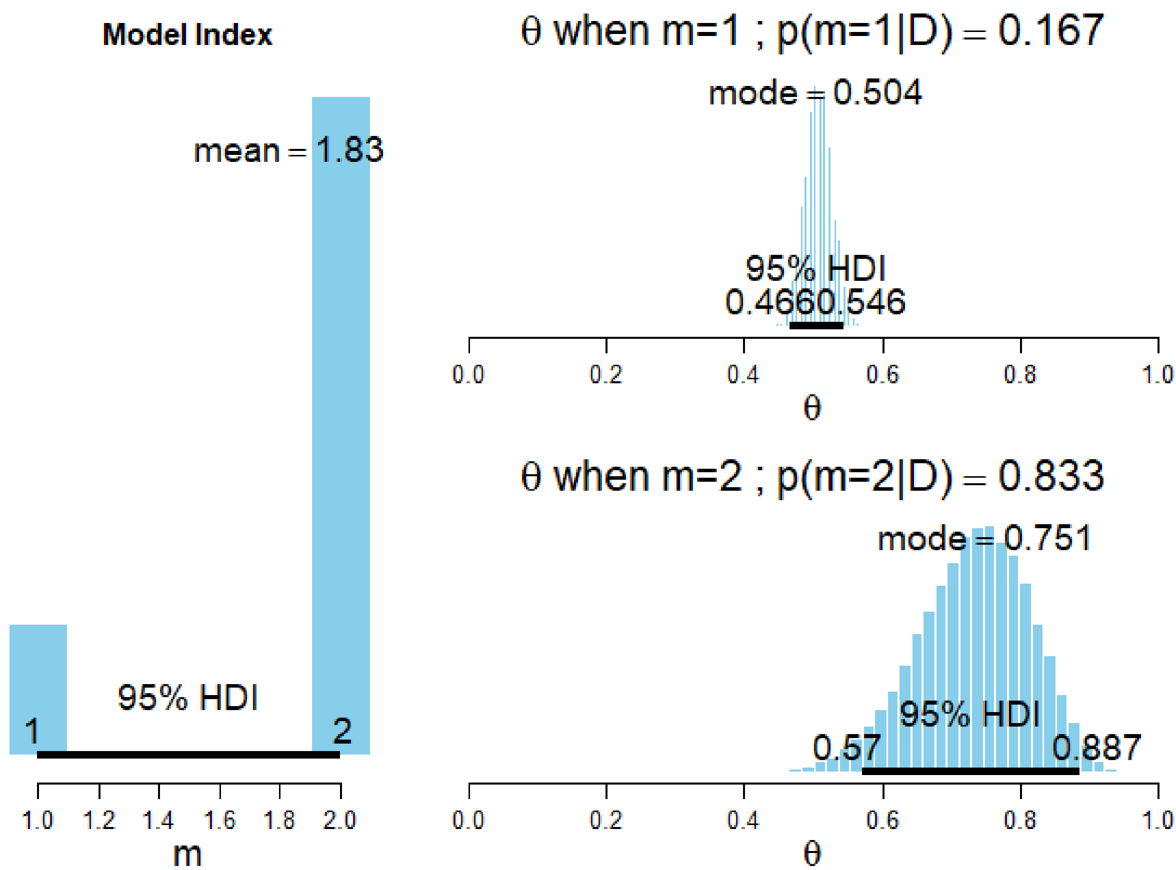

Figure 2:

```
## Graph information:
##    Observed stochastic nodes: 0
##    Unobserved stochastic nodes: 18
##    Total graph size: 39
##
## Initializing model
##
## Burning in the MCMC chain...
## Sampling final MCMC chain...
```

Model with index=2 is highly prefered over model 1 by 0.666. The 95% HDI on the theta value in the biased-factory model (i.e., in m=2) lies between 0.57 and 0.887. And It is evident from the above posterior that theta value of 0.5 is clearly excluded.

1C.

```
source("Jags-Ydich-Xnom1subj-MbernBetaModelComp.R")
```

```
##
## *********************************************************************
## Kruschke, J. K. (2015). Doing Bayesian Data Analysis, Second Edition:
## A Tutorial with R, JAGS, and Stan. Academic Press / Elsevier.
```

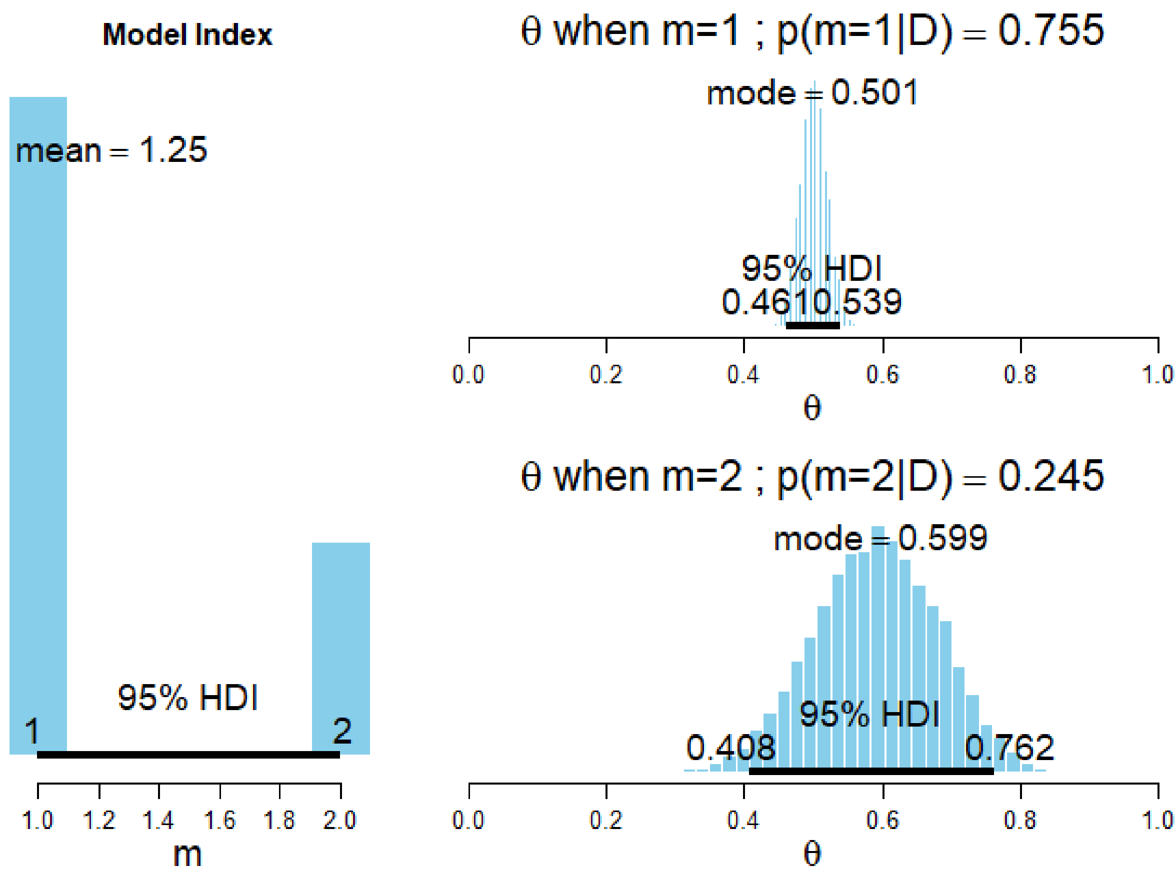

Figure 3:

```
## ***********************************************************************
##
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 0
##    Unobserved stochastic nodes: 18
##    Total graph size: 39
##
## Initializing model
##
## Burning in the MCMC chain...
## Sampling final MCMC chain...
```

Now with the new data (N=16 and z=8), model with index 1 is preferred over model 2 by 0.510. 95% HDI on the theta value in the biased-factory model (i.e., in m=2) lies between 0.408 and 0.762. It does include a theta value of 0.5 with reasonable credibility in fact.

2A.

```
# Data:
N = 45 ; z = 3
# p(y=1):
theta = 1/6
low_Tail= 0:z
# calculates Cumulative low tail probability given by equation 11.5 in the book:
lowTailProba = sum( choose(N,low_Tail) * theta^low_Tail * (1-theta)^(N-low_Tail) )
# Two tail probability:
TwoTailPval = 2 * lowTailProba
show( TwoTailPval )
```

```
## [1] 0.08920334
```

It considered the low tail and not the high tail because z/N = 3/45 is less than expected p = 1/6. We can not reject the hypothesis that theta=1/6 as the the two-tailed p value is 0.089, which is greater than .05.

2B.

```
complN = z:(N-1)
# calculates Cumulative low tail probability given by equation 11.6 in the book:
complP = sum( (z/complN) * choose(complN,z) * theta^z * (1-theta)^(complN-z) )
lowTailProba = 1-complP
# Two tail probability:
TwoTailPval = 2 * lowTailProba
show( TwoTailPval )
```

```
## [1] 0.03126273
```

Cumulative low tail probability is given by negative binomial. In this case, the tail is over n>=N, which can be easily computed by using the finite complement to its right i.e. 1 - p(n<N). Here we can reject the hypothesis that theta=1/6 because TwoTailProba is 0.0312 which is less than 0.05

3A.

```
N = 45 ; z = 3
theta = 1/6
# z/N = 3/45 = 0.066
# Computing the left-tail p value For each candidate theta values from 0.170 to 0.190 with a step size
lowTailZ = 0:z
for ( theta in seq( 0.170 , 0.190 , 0.001) ) {
  show( c(theta , 2*sum( choose(N,lowTailZ) * theta^lowTailZ * (1-theta)^(N-lowTailZ) ) ) )
}
```

```
## [1] 0.170000 0.079301
## [1] 0.17100000 0.07652818
## [1] 0.17200000 0.07384245
## [1] 0.17300000 0.07124149
## [1] 0.17400000 0.06872304
## [1] 0.17500000 0.06628487
## [1] 0.17600000 0.06392479
## [1] 0.17700000 0.06164067
```

```
## [1] 0.17800000 0.05943039
## [1] 0.17900000 0.05729192
## [1] 0.18000000 0.05522324
## [1] 0.18100000 0.05322237
## [1] 0.1820000 0.0512874
## [1] 0.18300000 0.04941643
## [1] 0.18400000 0.04760762
## [1] 0.18500000 0.04585918
## [1] 0.18600000 0.04416934
## [1] 0.18700000 0.04253637
## [1] 0.18800000 0.04095861
## [1] 0.1890000 0.0394344
## [1] 0.19000000 0.03796215
```

The highest theta value which can not be rejected is 0.182.

Similarly 2nd script computes the right-tail p value for candidate theta values from 0.005 to 0.020 with step size of 0.001, which are less than z/N observed.

```
# The columns are theta and p value:
# For candidate theta values from 0.005 to 0.020, which are less than z/N observed, # compute the right
highTailZ = z:N
for ( theta in seq( 0.005 , 0.020 , 0.001) ) {
show( c(
theta ,
2*sum( choose(N,highTailZ) * theta^highTailZ * (1-theta)^(N-highTailZ) )
))
}
```

```
## [1] 0.005000000 0.003032142
## [1] 0.006000000 0.005078266
## [1] 0.007000000 0.007816233
## [1] 0.00800000 0.01130928
## [1] 0.00900000 0.01560897
## [1] 0.01000000 0.02075627
## [1] 0.01100000 0.02678246
## [1] 0.01200000 0.03371014
## [1] 0.01300000 0.04155399
## [1] 0.0140000 0.0503216
## [1] 0.01500000 0.06001419
## [1] 0.01600000 0.07062725
## [1] 0.01700000 0.08215124
## [1] 0.01800000 0.09457208
## [1] 0.0190000 0.1078717
## [1] 0.0200000 0.1220287
```

```
# The columns are theta and p value:
```

The lowest theta that can not be rejected is 0.014.

Confidence Interval is in between 0.014 and 0.182. It is the range of theta values not rejected when the stopping intention is fixed N.

3B.

```r
# Compute the left tailed p value.
# For candidate theta values between 0.15 and 0.16 with step size of 0.001, greater than z/N observed,
complN = z:(N-1)
for ( theta in seq( 0.150 , 0.160 , 0.001) ) {
show( c(theta, 2*(1-sum( (z/complN) * choose(complN,z) * theta^z * (1-theta)^(complN-z) ) ) ) )
}
```

```
## [1] 0.15000000 0.05995277
## [1] 0.15100000 0.05770908
## [1] 0.15200000 0.05554267
## [1] 0.15300000 0.05345117
## [1] 0.15400000 0.05143228
## [1] 0.15500000 0.04948376
## [1] 0.15600000 0.04760341
## [1] 0.15700000 0.04578911
## [1] 0.15800000 0.04403879
## [1] 0.15900000 0.04235041
## [1] 0.16000000 0.04072202
```

The highest theta value which can not be rejected is 0.154.

```r
# For candidate theta values LESS than z/N observed, compute the RIGHT-tail p
# value: >
highTailN = z:N # Notice N not N-1
for ( theta in seq( 0.005 , 0.020 , 0.001) ) {
 show( c(theta , 2*sum( (z/highTailN) * choose(highTailN,z) * theta^z * (1-theta)^(highTailN-z) ) ) )
}
```

```
## [1] 0.005000000 0.003032142
## [1] 0.006000000 0.005078266
## [1] 0.007000000 0.007816233
## [1] 0.00800000 0.01130928
## [1] 0.00900000 0.01560897
## [1] 0.01000000 0.02075627
## [1] 0.01100000 0.02678246
## [1] 0.01200000 0.03371014
## [1] 0.01300000 0.04155399
## [1] 0.0140000 0.0503216
## [1] 0.01500000 0.06001419
## [1] 0.01600000 0.07062725
## [1] 0.01700000 0.08215124
## [1] 0.01800000 0.09457208
## [1] 0.0190000 0.1078717
## [1] 0.0200000 0.1220287
```

The lowest theta value which can not be rejected is 0.14.

Thus, when the stopping at fixed z, the confidence interval goes from 0.014 to 0.154. Confidence Interval is very different from previous scenario where we stopped at fixed N.

4.

```r
# Given Data:
N = 45 ; z = 3
# Hypothetical value of parameter theta
theta = 1/6
# Fixed duration rolling of N=40 to 50 with equal probability
Nposs = 40:50
# Specify probability of each N (all with equal probability):
Nprob = rep(1, length(Nposs))
# Normalized to get actual probability mass.
Nprob = Nprob/sum(Nprob)
# Initialize total tail probability to zero.
totalP = 0
# For each N, computing the p value, and accumulate the weighted total p:
for ( i in 1:length(Nposs) )
{
thisN = Nposs[i]
# For this N, determine the max z that is in the low tail.
# It must satisfy thisZ/thisN <= z/N.
thisZ = max( (0:thisN)[ (0:thisN)/thisN <= z/N ] )
# Sum of binomial probabilities from z/N to zero.
lowTailZ = 0:thisZ
thisP = 2*sum(choose(thisN,lowTailZ) * theta^lowTailZ * (1-theta)^(thisN-lowTailZ) )
# Calculating the weighted average now
totalP = totalP + Nprob[i] * thisP
# Print P value during the fixed duration:
show( c( thisN , thisP ) )
}
```

```
## [1] 40.00000000  0.05470238
## [1] 41.00000000  0.04762645
## [1] 42.00000000  0.04142745
## [1] 43.00000000  0.03600333
## [1] 44.00000000  0.03126273
## [1] 45.00000000  0.08920334
## [1] 46.00000000  0.07885681
## [1] 47.00000000  0.06963315
## [1] 48.0000000  0.0614227
## [1] 49.00000000  0.05412451
## [1] 50.00000000  0.04764606
```

```r
show( totalP )
```

```
## [1] 0.05562808
```

P value for the observed outcome is 0.055 This p value is not the same as that of fixed N or fixed Z.

Above script caluclates the average of P value for each possible N weighted by probability that that would happen.

5.

```r
# Given Data:
N = 45 ; z = 3
# Hypothetical value of parameter theta
theta = 1/6
# Fixed duration rolling of N=40 to 50 with equal probability
Nposs = 35:45
# Specify probability of each N (all with equal probability):
Nprob = rep(1, length(Nposs))
# Normalized to get actual probability mass.
Nprob = Nprob/sum(Nprob)
# Initialize total tail probability to zero.
totalP = 0
# For each N, computing the p value, and accumulate the weighted total p:
for ( i in 1:length(Nposs) )
{
thisN = Nposs[i]
# For this N, determine the max z that is in the low tail.
# It must satisfy thisZ/thisN <= z/N.
thisZ = max( (0:thisN)[ (0:thisN)/thisN <= z/N ] )
# Sum of binomial probabilities from z/N to zero.
lowTailZ = 0:thisZ
thisP = 2*sum(choose(thisN,lowTailZ) * theta^lowTailZ * (1-theta)^(thisN-lowTailZ) )
# Calculating the weighted average now
totalP = totalP + Nprob[i] * thisP
# Print P value during the fixed duration:
show( c( thisN , thisP ) )
}
```

```
## [1] 35.0000000  0.1076747
## [1] 36.00000000  0.09424354
## [1] 37.00000000  0.08239256
## [1] 38.00000000  0.07195241
## [1] 39.00000000  0.06276894
## [1] 40.00000000  0.05470238
## [1] 41.00000000  0.04762645
## [1] 42.00000000  0.04142745
## [1] 43.00000000  0.03600333
## [1] 44.00000000  0.03126273
## [1] 45.00000000  0.08920334
```

```r
show( totalP )
```

```
## [1] 0.06538707
```

P value depends on the sampling scheme when sampling for a fixed duration. When the roller could have rolled N=35, N=36, N=37, ., N=45 with equal probabilities, P value changed to 0.065.