HW03.Rmd

Shashi

January 29, 2018

R Markdown

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

1.

```
source("DBDA2E-utilities.R") # Load definitions of graphics functions etc.
```

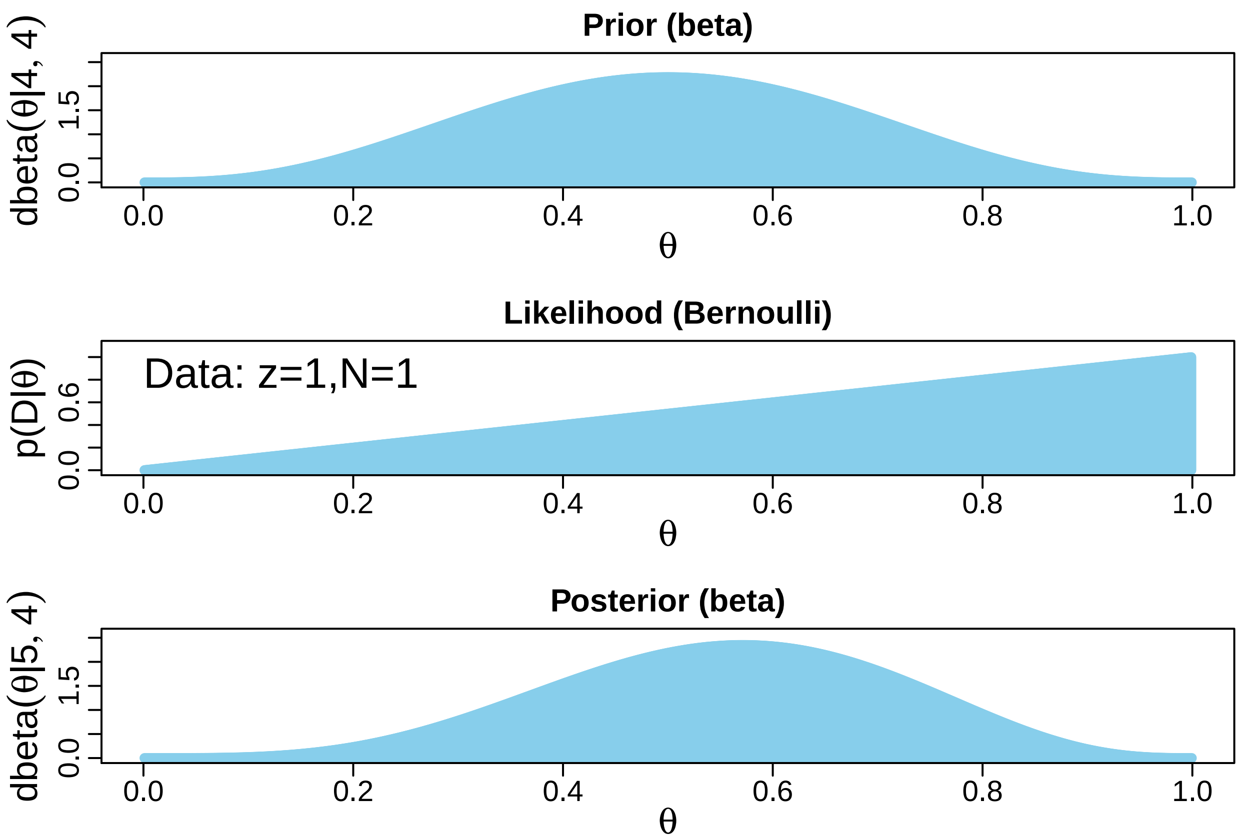
```
##
## *******************************************************************
## Kruschke, J. K. (2015). Doing Bayesian Data Analysis, Second Edition:
## A Tutorial with R, JAGS, and Stan. Academic Press / Elsevier.
## *******************************************************************
```

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

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

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

```
source("BernBeta.R") # Load the definition of the BernBeta function
openGraph()
post = BernBeta(priorBetaAB=c(4,4), Data=c(1))
```

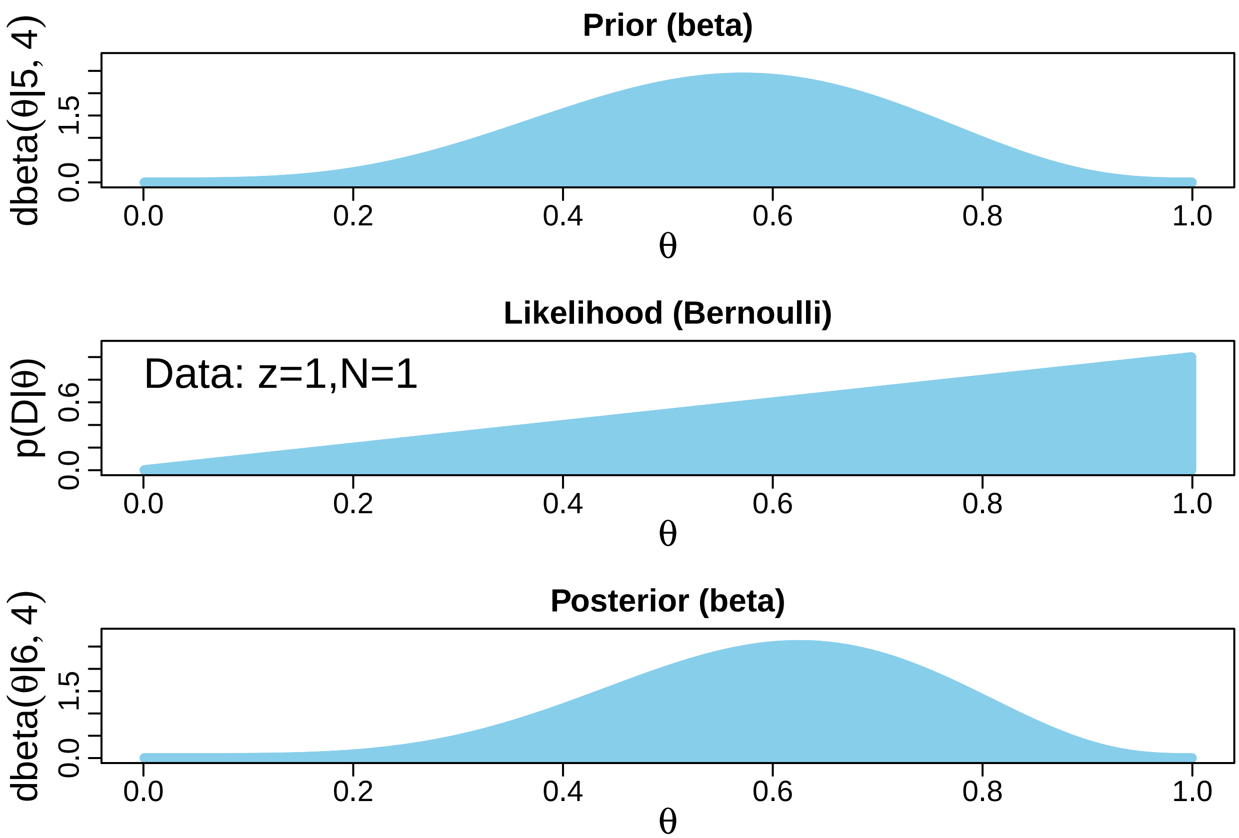



```
show(post)
```

```
## [1] 5 4
```

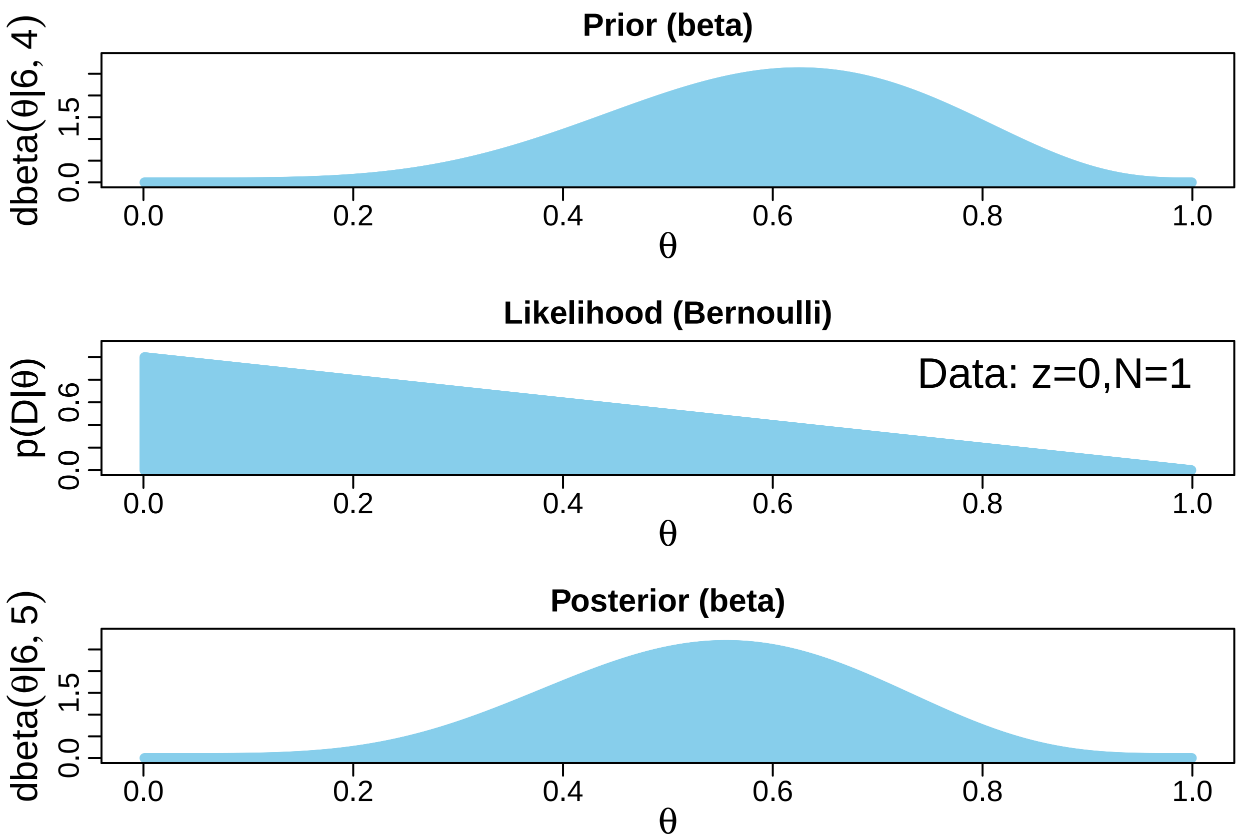
The posterior is dbeta(theta| 5,4).

```
# Using the posteriror from previous step as prior for this step
post = BernBeta(priorBetaAB = post, Data = c(1))
```



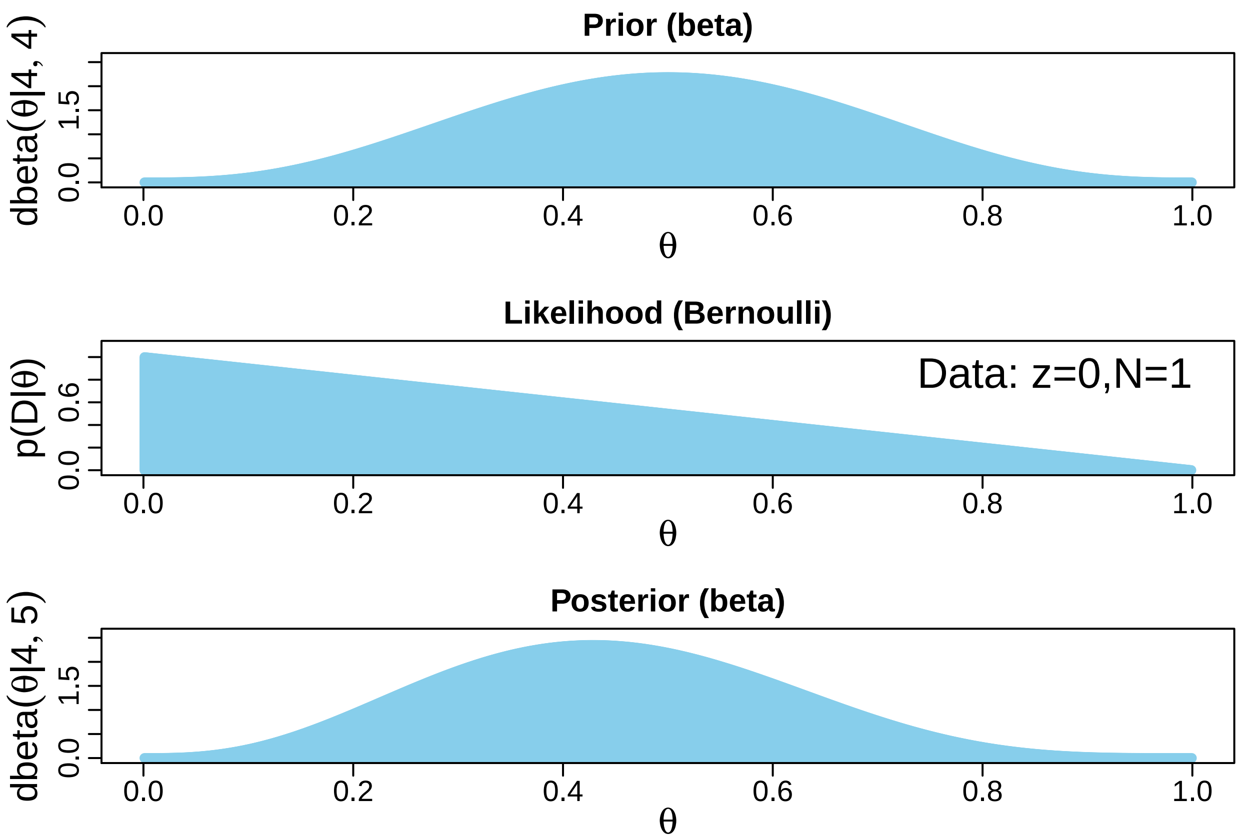
The posteriror is dbeta(theta| 6,4).

```
post = BernBeta(post, Data = c(0))
```

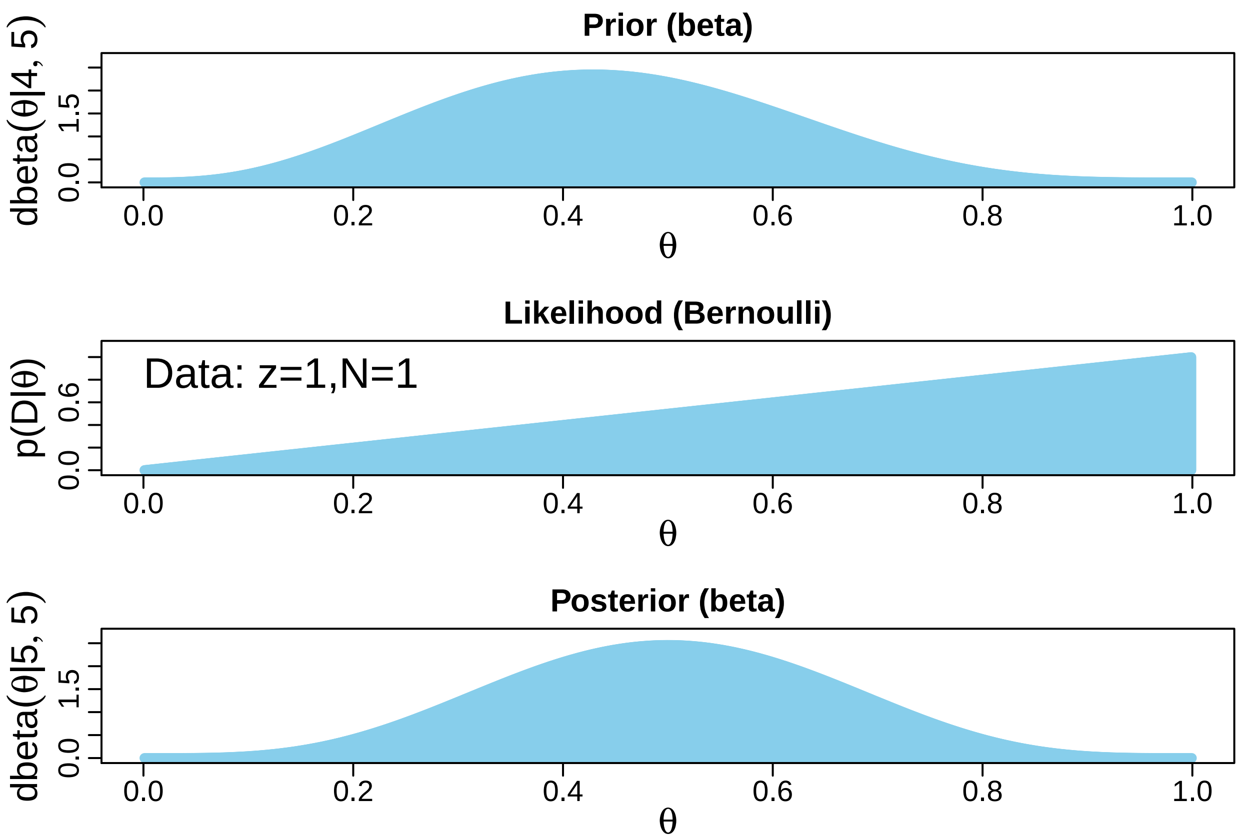


The posterior is dbeta(theta| 6,5).

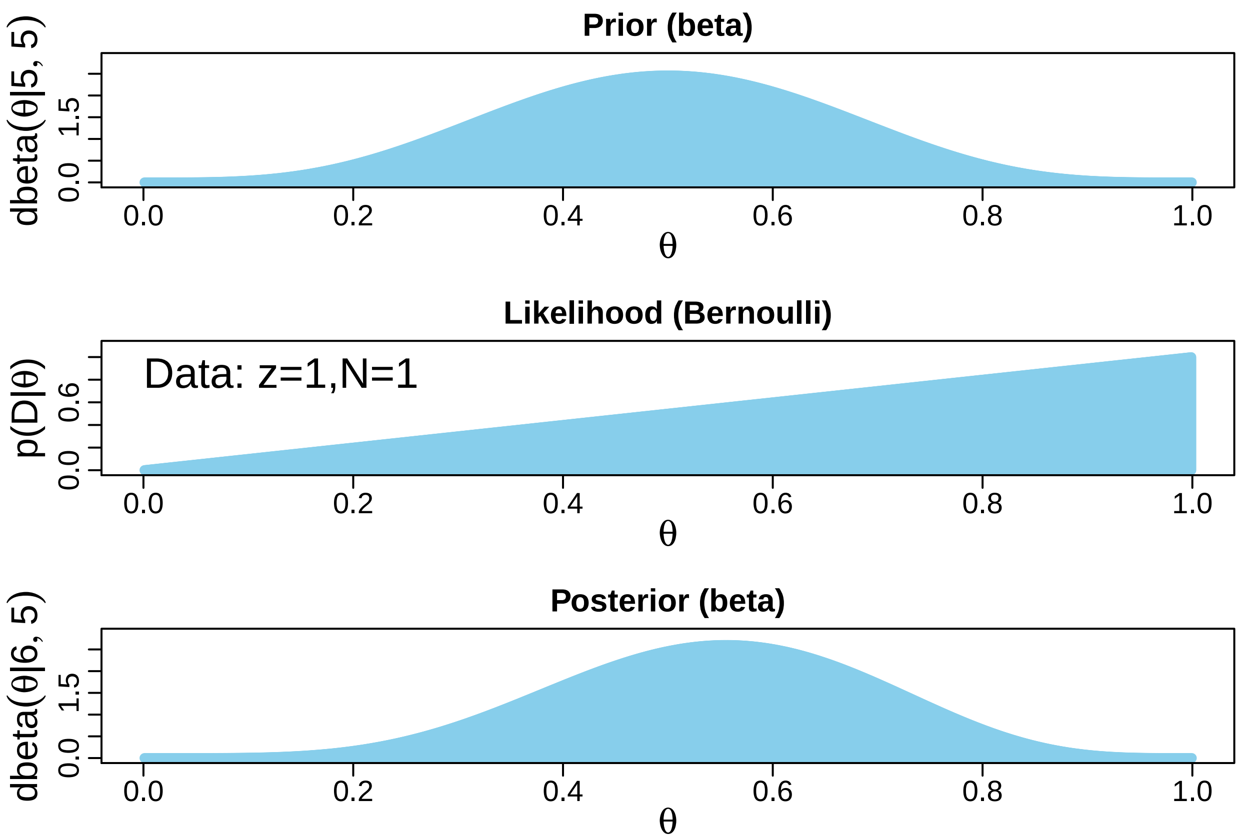
```
post = BernBeta(priorBetaAB=c(4,4), Data=c(0))
```




```
post = BernBeta(priorBetaAB=post, Data=c(1))
```



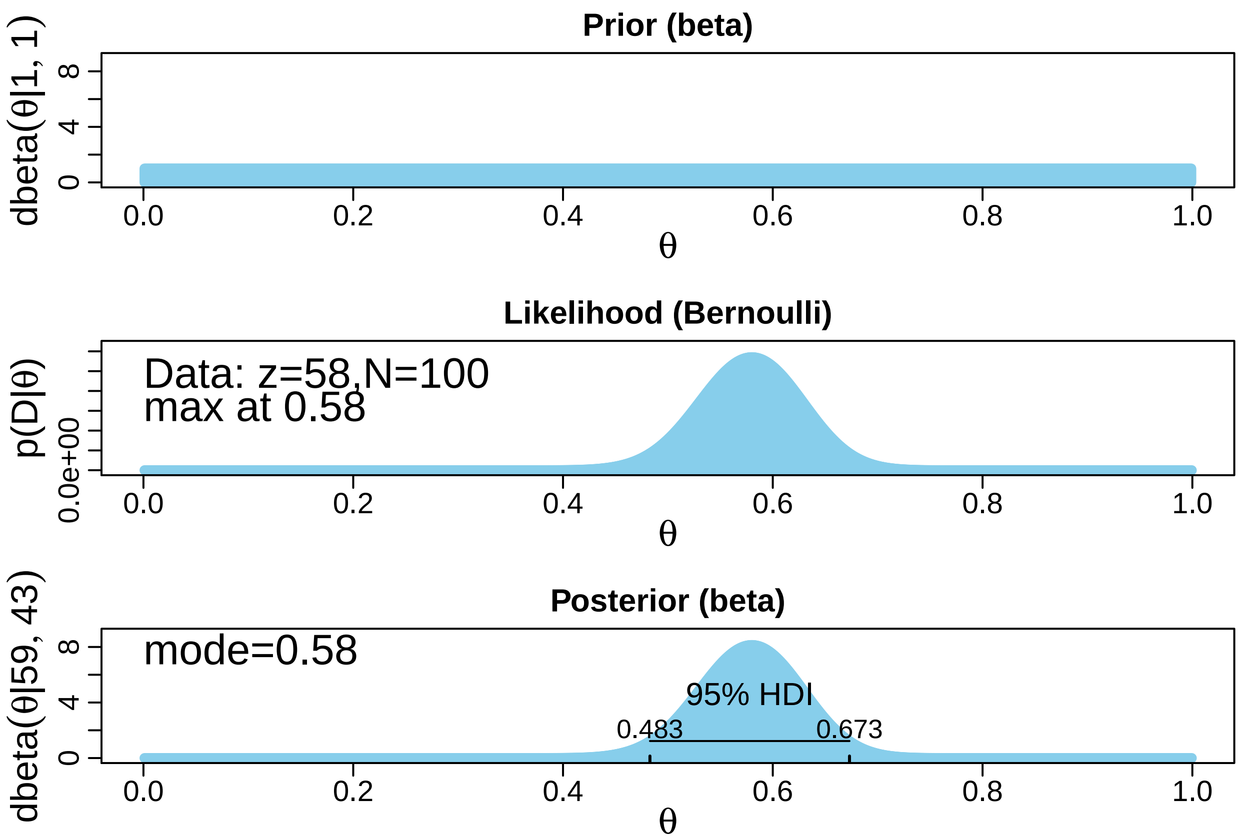

```
post = BernBeta(priorBetaAB=post, Data=c(1))
```



The final posterior is still the same dbeta(theta| 6,5).

2A.

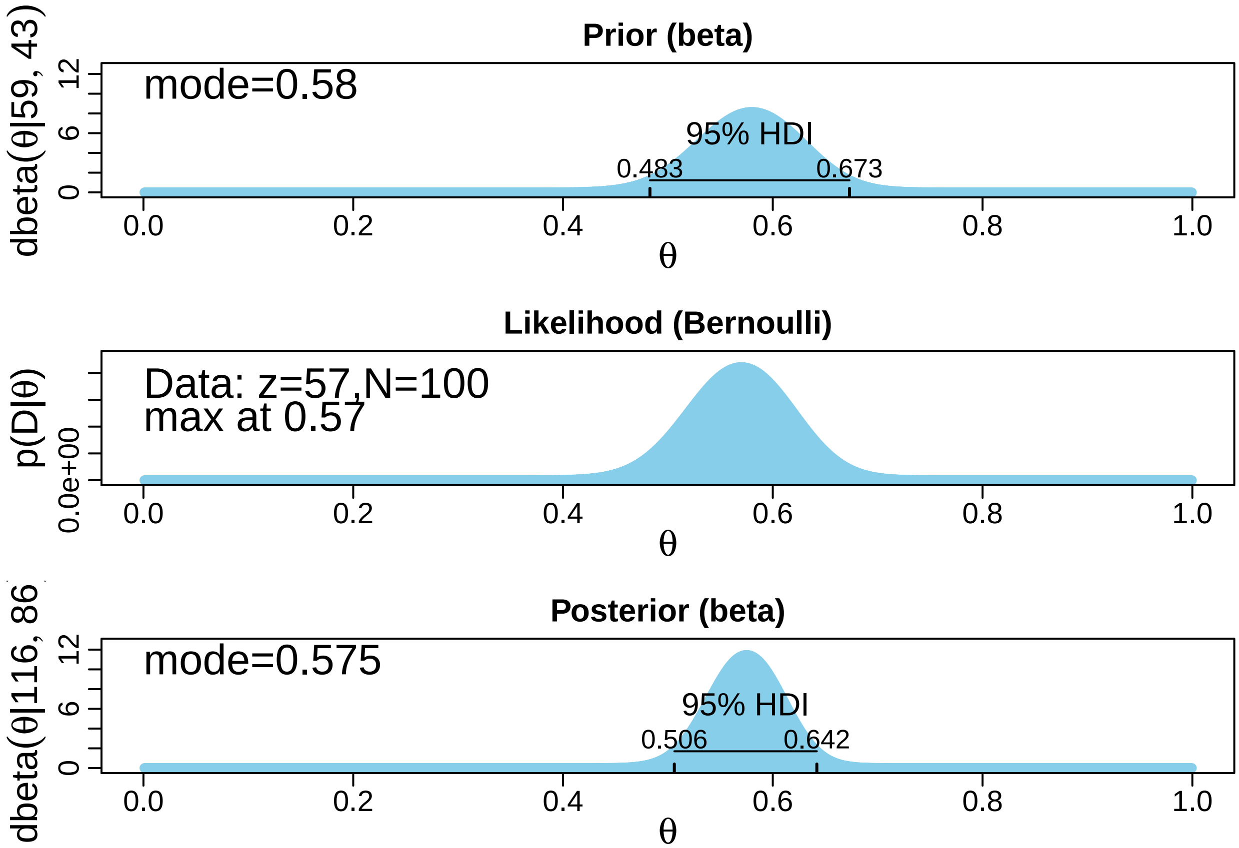
```
# prior belief is uniform ditribution
post = BernBeta(priorBetaAB = c(1,1), Data = c(rep(1,58), rep(0,100-58)),
showHDI=TRUE, showCentTend="Mode")
```



For probability of candidate A, 95% HDI lies between 0.483 and 0.673.

2B.

```
# prior belief is the posterior from the previous experiment now.
post = BernBeta(priorBetaAB = post, Data = c(rep(1,57), rep(0,100-57)),
showHDI=TRUE, showCentTend="Mode")
```



For probability of candidate A, 95% HDI lies between 0.506 and 0.642.

3A.

```r
source("DBDA2E-utilities.R")
```

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

```r
mode = 2/3 #two-thirds of them survived at least one year after surgery
kappa = 50
betaparams = betaABfromModeKappa( mode=mode , kappa=kappa )
betaparams$a
```

```
## [1] 33
```

```r
betaparams$b
```

```
## [1] 17
```

Parameter a is 33 and parameter b is 17.

3B.

```r
source("DBDA2E-utilities.R")
```

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

```r
mode = 1/6 #two-thirds of them survived at least one year after surgery
kappa = 50
betaparams = betaABfromModeKappa( mode=mode , kappa=kappa )
betaparams$a
```

```
## [1] 9
```

```r
betaparams$b
```

```
## [1] 41
```

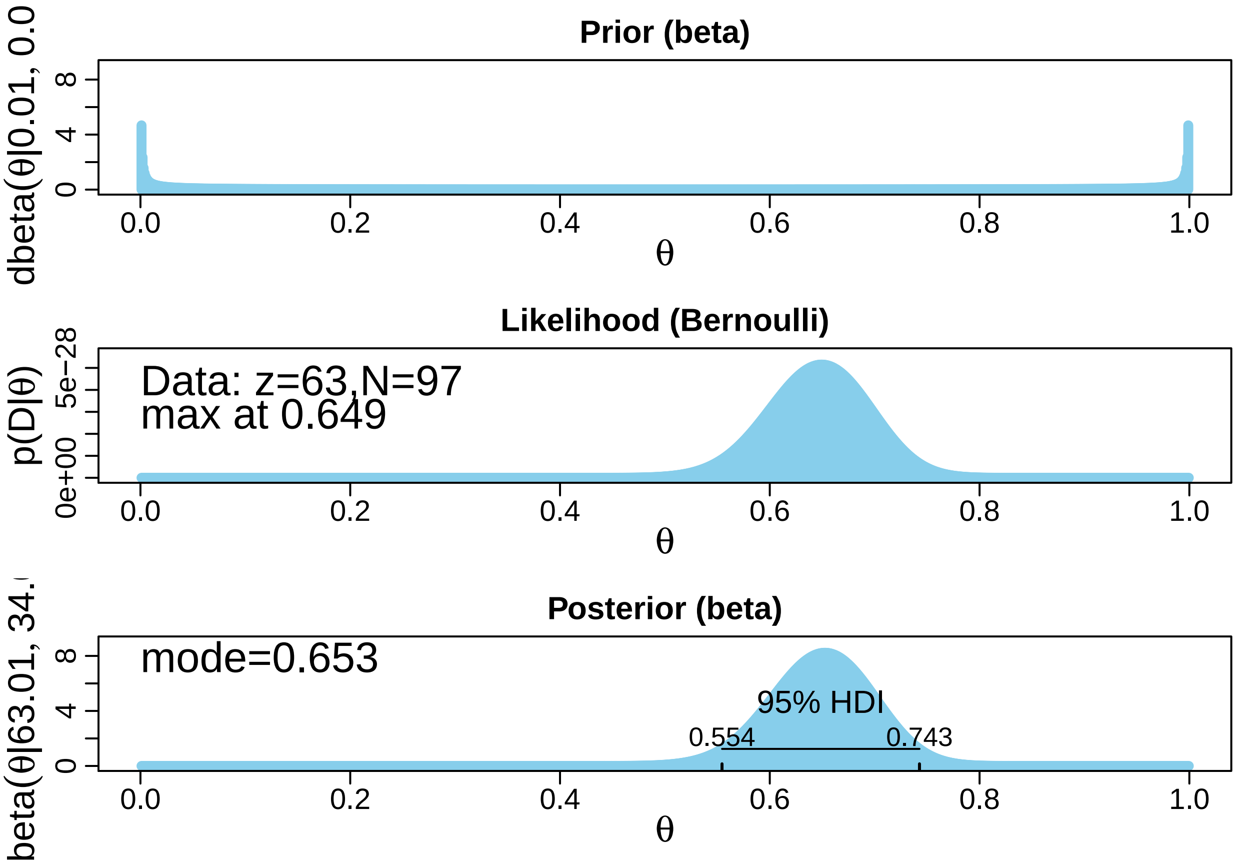
Parameter a is 9 and parameter b is 41.

4A.

```r
setwd("C://Users//hoosi//Desktop//BDA//DBDA2Eprograms")
source("DBDA2E-utilities.R") # Load definitions of graphics functions etc.
```

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

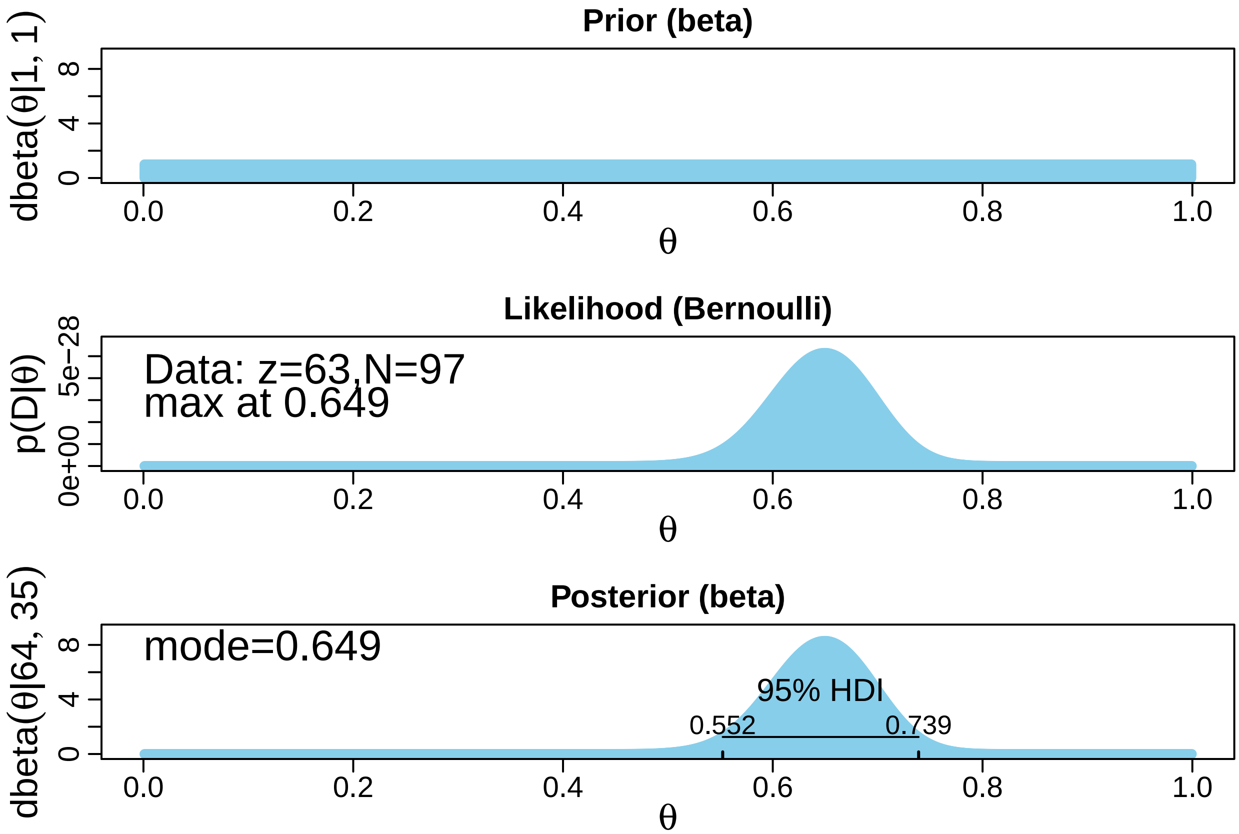
```r
source("BernBeta.R") # Load the definition of the BernBeta function
#openGraph()
post = BernBeta(priorBetaAB = c(0.01, 0.01), Data = c(rep(1,63), rep(0, 97-63)),
showHDI=TRUE, showCentTend="Mode")
```



Mode is 0.653 and 95% HDI of the posterior lies between 0.554 and 0.743

4B.

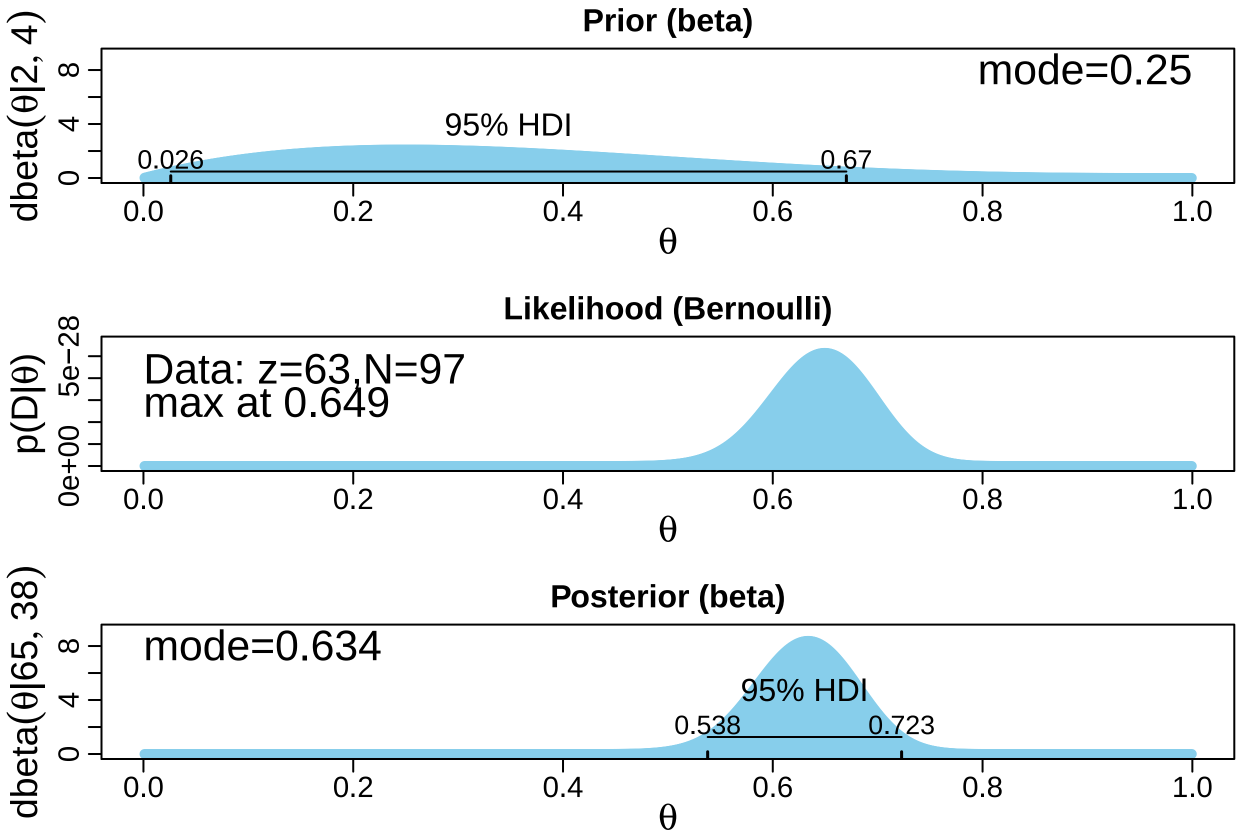
```
post = BernBeta(priorBetaAB = c(1, 1), Data = c(rep(1,63), rep(0, 97-63)),
showHDI=TRUE, showCentTend="Mode")
```



Mode is 0.649 and 95% HDI of the posterior lies between 0.552 and 0.739

4C.

```
post = BernBeta(priorBetaAB = c(2, 4), Data = c(rep(1,63), rep(0, 97-63)),
showHDI=TRUE, showCentTend="Mode")
```



Mode is 0.634 and 95% HDI of the posterior lies between 0.538 and 0.723

4D.

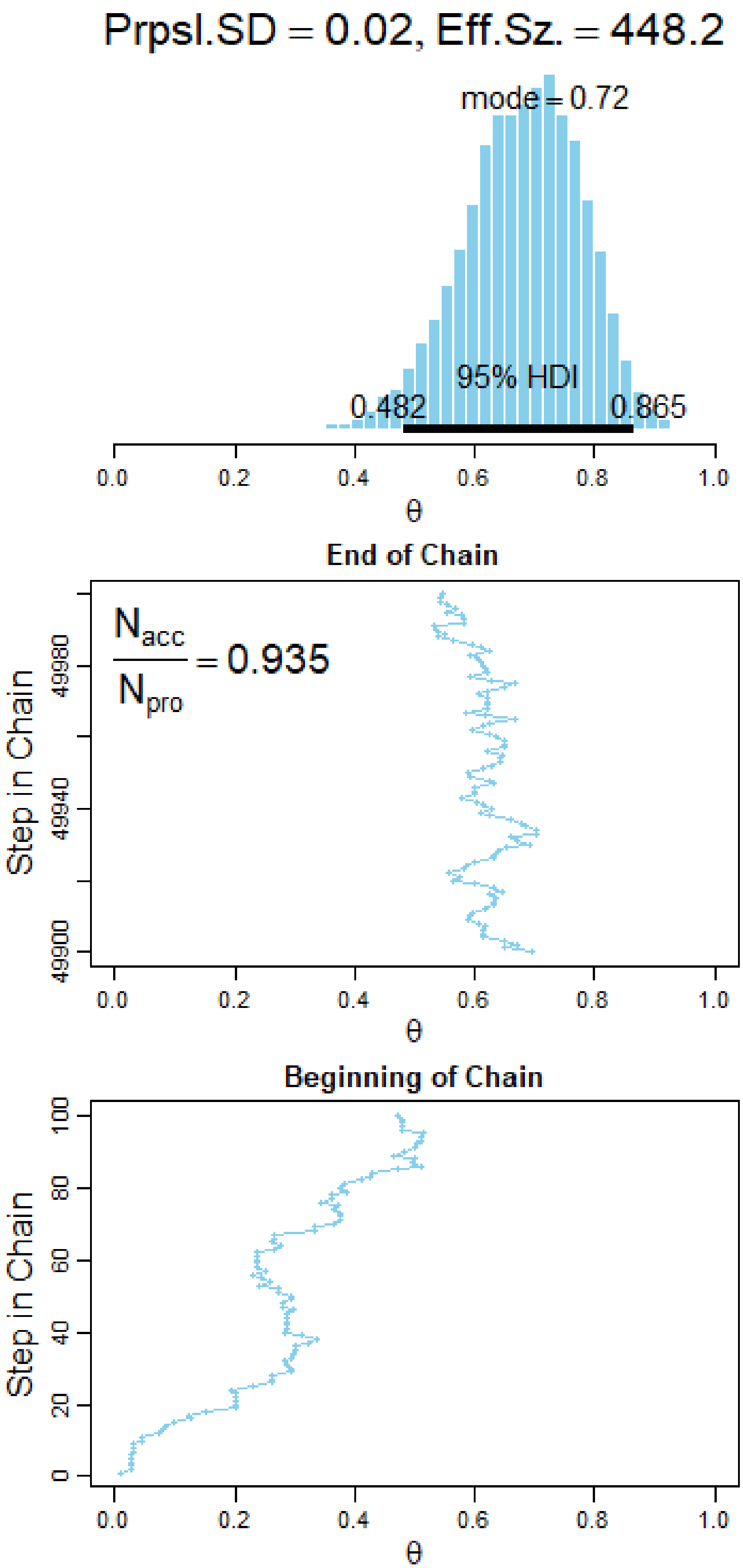
There isn't much difference in mode and HDIs of posterior distributions across the different priors.

```r
setwd("C://Users//hoosi//Desktop//BDA//DBDA2Eprograms")
source("DBDA2E-utilities.R") # Load definitions of graphics functions etc.
```

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

```r
source("BernMetrop.R")
```

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



Prpsl.SD = 0.02, Eff.Sz. = 448.2
mode = 0.72
95% HDI
0.482
0.865
0.0
0.2
0.4
0.6
0.8
1.0
θ
End of Chain
$\frac{N_{acc}}{N_{pro}} = 0.935$
Step in Chain
49900
49940
49980
Beginning of Chain
0
20
40
60
80
100

Figure 1:

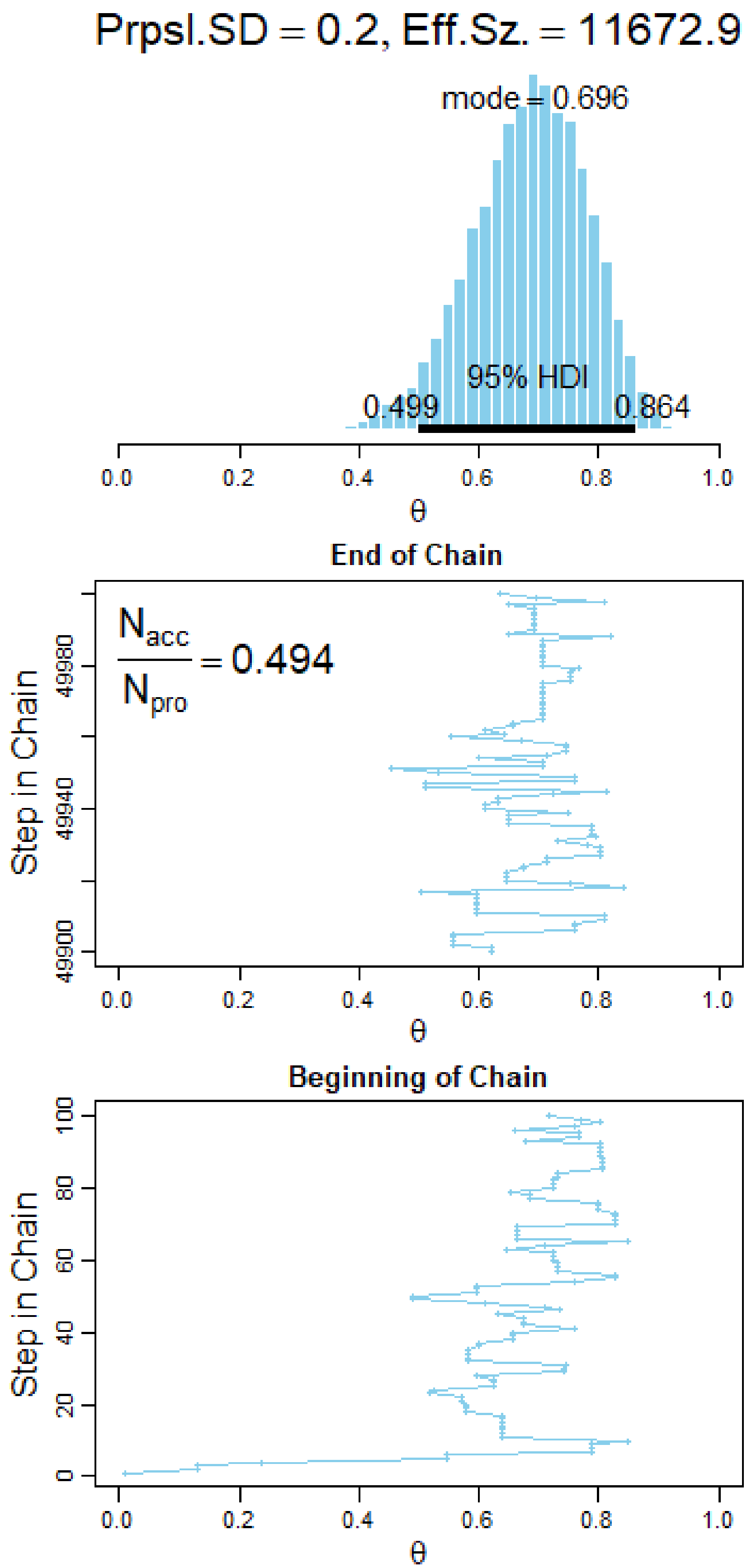
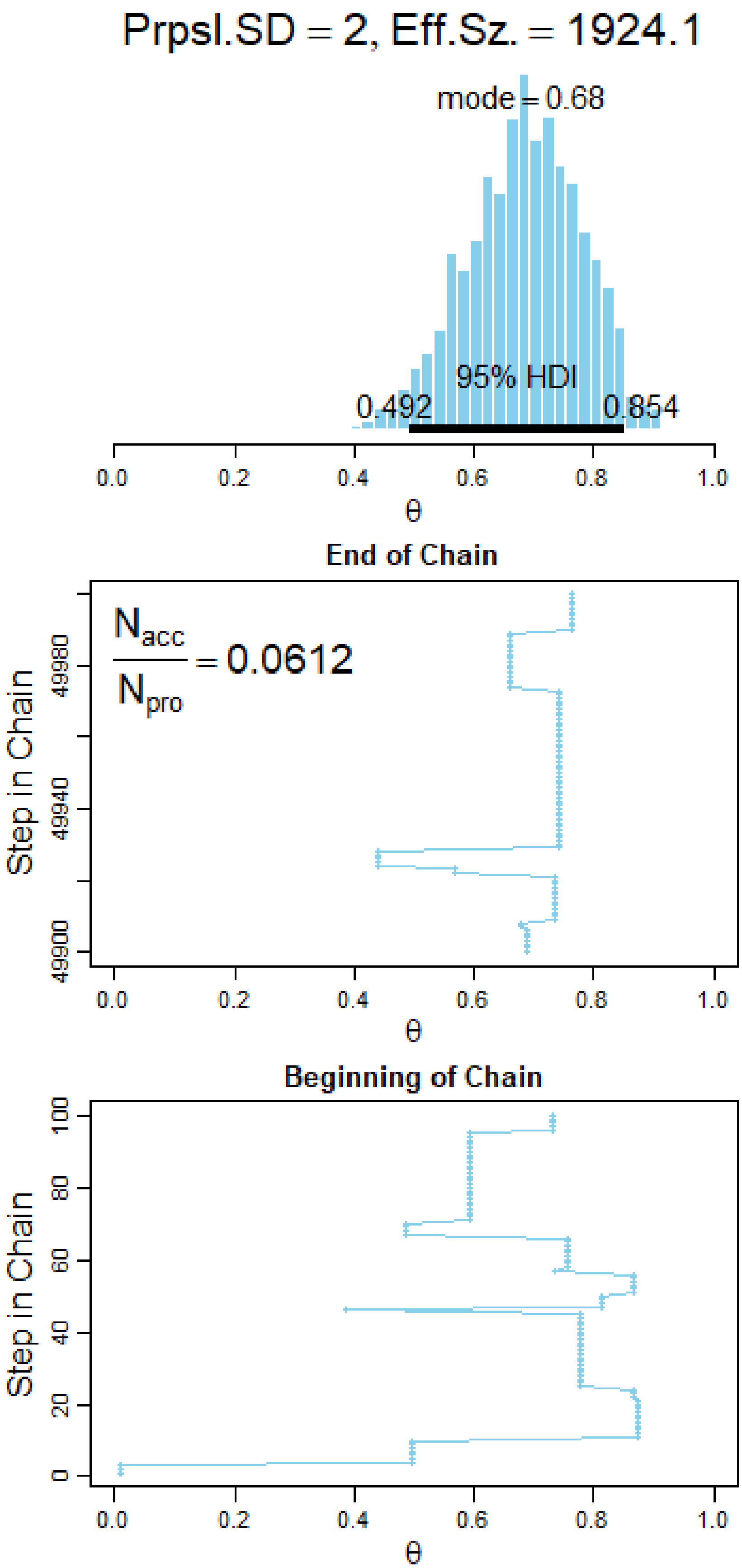

Prpsl.SD = 0.2, Eff.Sz. = 11672.9
mode = 0.696
95% HDI
0.499
0.864
θ
End of Chain
Step in Chain
$\frac{N_{acc}}{N_{pro}} = 0.494$
θ
Beginning of Chain
Step in Chain
θ

Figure 2:



Commented out the seed. The

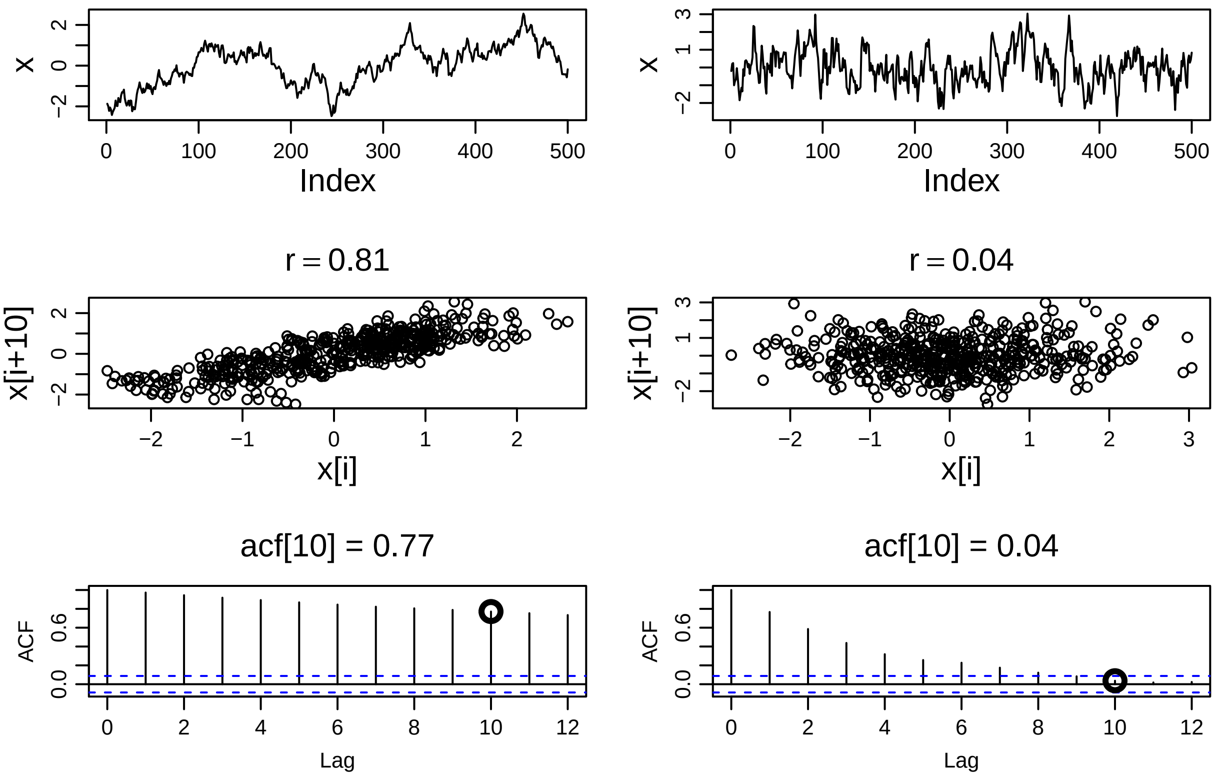
graphs showed similar behavior to those in Figure 7.4: The acceptance ratio was the highest for proposal SD=0.02, moderate for SD=0.2, and very low for SD=2.0. The ESS is best for proposal SD=0.2(11672.9), but poor for SD=0.02 and for SD=2.0.

6A.

```
# first two lines are optional:
source("DBDA2E-utilities.R")
```

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

```
openGraph(width=5,height=7.5) #open a graphics window
# SSpecifying the location of 6 plots
layout( matrix(1:6,nrow=3) )
par(mar=c(4,4,3,1),mgp=c(2,0.7,0)) # Setting the margin size (mar), axis label locations (mgp)
maxLag=10
for ( inert in c(0.99,0.80) ) { # Iterate over inertia values
x = rep(0,500) # create a vector with 500 zeroes in it
for ( i in 2:length(x) ) {
x[i] = inert*x[i-1] + rnorm(1,0,1-inert) # Add noise to data points from a normal distribution with mea
}
x = (x-mean(x))/sd(x)
# trace plot of data points against index
plot( x , ylab="x" , cex.lab=1.5 , type="l" )
# Scatterplot with given lag size
plot( x[1:(500-maxLag)] , x[(maxLag+1):500] ,
xlab=bquote("x[i]") , ylab=bquote("x[i+"*.(maxLag)*"]") ,
cex.lab=1.5 , cex.main=1.5 ,
main=bquote(r==.(round(cor(x[1:(500-maxLag)],x[(maxLag+1):500]),2))) )
# Compute autocorrelation function of accepted trjectory, with given lags and plot the function:
acfInfo = acf( x , lag.max=maxLag+2 , main="" )
title( main=bquote("acf["*.(maxLag)*"] = "*.(round(acfInfo$acf[maxLag+1],2)) ) ,
cex.main=1.5 )
points( maxLag , acfInfo$acf[maxLag+1] , cex=2 , lwd=3 )
}
```



6B.

6C.Scatterplots in the 2nd row show the MCMC trjectory plotted against its values 10 steps later. If there is less movement and occasional big jumps in the trace plot that means it has high autocorrelation value (in column 1). On the other hand, column 2 trace plot has usually sharp jumps and thus low autocorrelation.

6D.Bar graphs in the third row are showing autocorrelation with respect to different values of lags. The circle in the bar plot at Lag=10 has a height of r1 and r2 in the col1 and col2 plots respectively, which are matched by the autocorrelation values in the title of the right scatter plots above.

6E.Column 1 data is highly autocorrelated and column2 has low autocorrelation.

6F. Column 2 has low autocorrelation as we can see that each step moves very far from its previous position most of the time, and usually has big jumps, Hence it will have larger ESS.