# Bayesian Stats HW4

[redacted]

```
graphics.off()
setwd("/Users/[redacted]Bayesian Data Analysis/HW")
source("/Users/[redacted]/Bayesian Data Analysis/DBDA2Eprograms/DBDA2E-utilities.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 4.3.0
```

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

```
source("/Users/[redacted]/Bayesian Data Analysis/DBDA2Eprograms/BernGrid.R")
```

## Ex 5.1

Suppose that the same randomly selected person as in Table 5.4 gets re-tested after the first test result was positive, and on the re-test, the result is negative. When taking into account the results of both tests, what is the probability that the person has the disease?

```
# probability of having disease
prob.dis = .001
# hit rate is .99
prob.pos.given.dis = .99
# false alarm is .05
prob.pos.given.nodis = .05
prob.nodis = 1 - prob.dis
prob.neg.given.dis = 1-prob.pos.given.dis
prob.neg.given.nodis = 1-prob.pos.given.nodis
# first test
prob.dis.given.pos = prob.pos.given.dis * prob.dis/
  (prob.dis * prob.pos.given.dis
   + prob.nodis * prob.pos.given.nodis)
print(paste("only considering the first test result, the probability is :", prob.dis.given.pos, sep = " "))
```

```
## [1] "only considering the first test result, the probability is : 0.019434628975265"
```

```
# for retest
prob.dis.retest = prob.dis.given.pos
prob.dis.given.neg.retest = prob.neg.given.dis * prob.dis.retest/
  (prob.dis.retest * prob.neg.given.dis +
     (1 - prob.dis.retest) * prob.neg.given.nodis)
print(paste("considering two test results, the probability is",
            prob.dis.given.neg.retest, sep = " "))
```

```
## [1] "considering two test results, the probability is 0.000208586165048544"
```

## Ex 5.3

(A) Suppose that a person selected at random from the population gets the test and it comes back negative. Compute the probability that the person has the disease.

```r
# probability of having disease (prior)
prob.dis = .001
# hit rate is .99
prob.pos.given.dis = .99
# false alarm is .05
prob.pos.given.nodis = .05
prob.nodis = 1 - prob.dis
prob.neg.given.dis = 1-prob.pos.given.dis
prob.neg.given.nodis = 1-prob.pos.given.nodis
# first test
prob.dis.given.neg = prob.neg.given.dis * prob.dis/
  (prob.dis * prob.neg.given.dis +
     prob.nodis * prob.neg.given.nodis)
print(paste("only considering the first test result, the probability is :",
            prob.dis.given.neg, sep = " "))
```

```
## [1] "only considering the first test result, the probability is : 1.05367416180221e-05"
```

(B) The person then gets re-tested, and on the second test the result is positive. Compute the probability that the person has the disease.

```r
prob.dis.retest = prob.dis.given.neg
prob.dis.given.pos.retest = prob.pos.given.dis * prob.dis.retest/
  (prob.dis.retest * prob.pos.given.dis +
     (1-prob.dis.retest) * prob.pos.given.nodis)
print(paste("only considering the first test result, the probability is :",
            prob.dis.given.pos.retest, sep = " "))
```

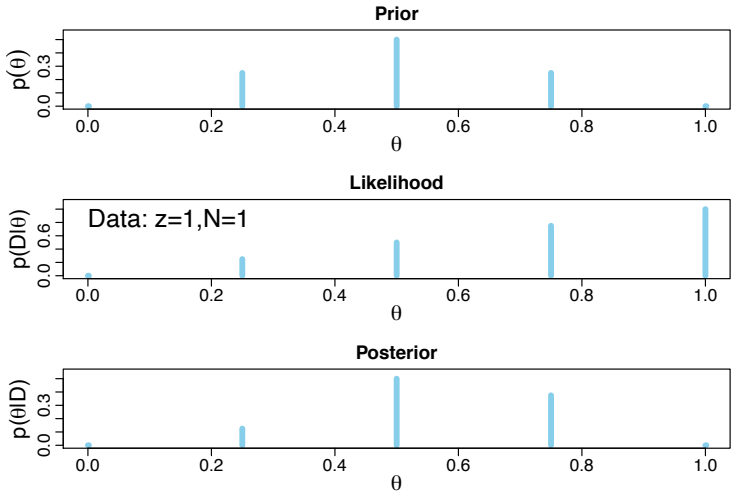
```
## [1] "only considering the first test result, the probability is : 0.000208586165048544"
```

The probability here is the same as what I got for Exercise 5.1

## Ex 5.4

**Example0:** this shows reallocation of 5 parameter values after taking into account prior and likelihood (because the likelihood favors Head). The prior has more influence because of the small sample size.

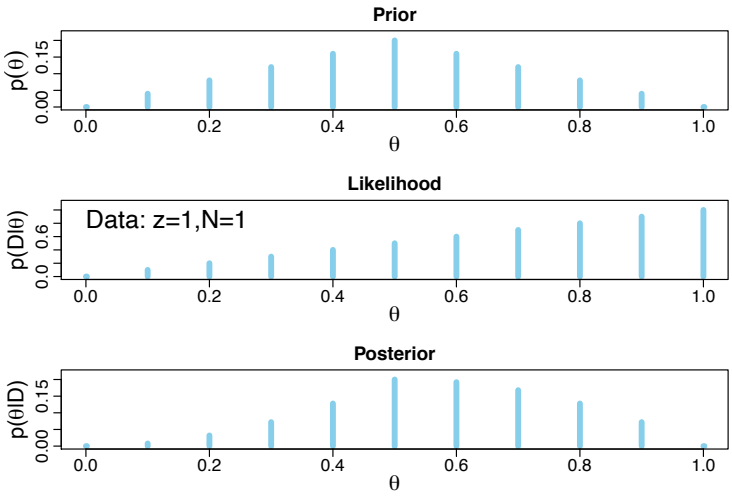
```r
Theta = seq( 0 , 1 , length=5 )  # Sparse teeth for Theta.
pTheta = pmin( Theta , 1-Theta ) # Triangular shape for pTheta.
pTheta = pTheta/sum(pTheta)      # Make pTheta sum to 1.0
Data = c(rep(0,0),rep(1,1))      # Single flip with 1 head
# openGraph(width=5,height=7)
posterior = BernGrid( Theta, pTheta , Data , plotType="Bars" ,
                      showCentTend="None", showHDI=FALSE, showpD=FALSE)
```



```
# saveGraph(file="BernGridExample0",type="eps")
```

**Example 1:** this shows reallocation of the distribution of 11 parameter values after taking into account prior and likelihood (because the likelihood favors Head). The prior has more influence because of the small sample size.

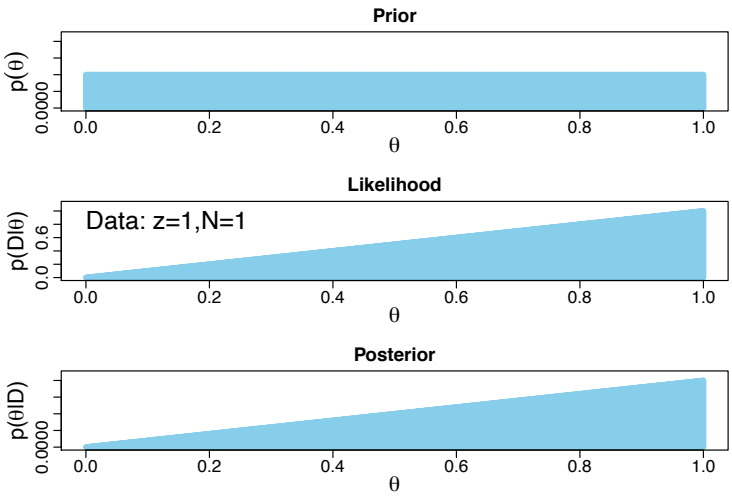
```
Theta = seq( 0 , 1 , length=11 )  # Sparse teeth for Theta.
pTheta = pmin( Theta , 1-Theta ) # Triangular shape for pTheta.
pTheta = pTheta/sum(pTheta)      # Make pTheta sum to 1.0
Data = c(rep(0,0),rep(1,1))      # Single flip with 1 head

# openGraph(width=5,height=7)
posterior = BernGrid( Theta, pTheta , Data , plotType="Bars" ,
                      showCentTend="None" , showHDI=FALSE , showpD=FALSE )
```




```
# saveGraph(file="BernGridExample1",type="eps")
```

**Example 2:** the shape of the posterior distribution is similar to that of the likelihood distribution because prior is flat (uninformed and vague) even though the sample size is super small

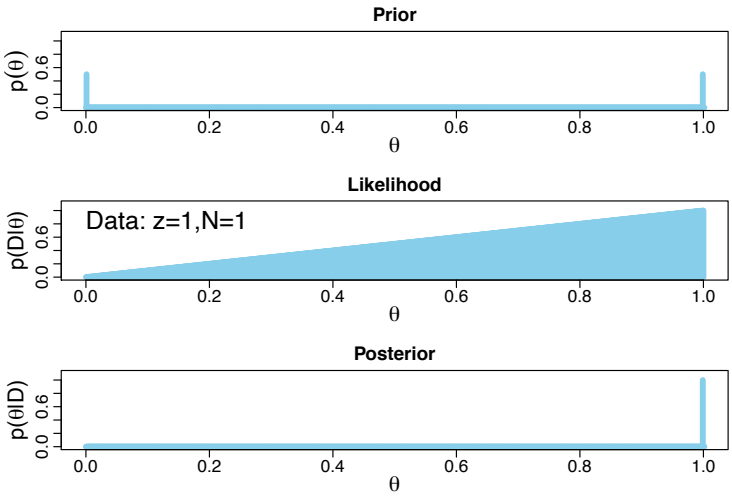
```
Theta = seq( 0 , 1 , length=1001 ) # Fine teeth for Theta.
pTheta = rep(1,length(Theta))      # Uniform (horizontal) shape for pTheta.
pTheta = pTheta/sum(pTheta)        # Make pTheta sum to 1.0
Data = c(rep(0,0),rep(1,1))        # Single flip with 1 head

# openGraph(width=5,height=7)
posterior = BernGrid( Theta, pTheta , Data , plotType="Bars" ,
                      showCentTend="None" , showHDI=FALSE , showpD=FALSE )
```




```
# saveGraph(file="BernGridExample2",type="eps")
```

**Example 3:** this shows an extreme distribution of priors and the likelihood favors head (only 1 flip). The posterior distribution favors prior distribution because it is informative and the sample size is super small

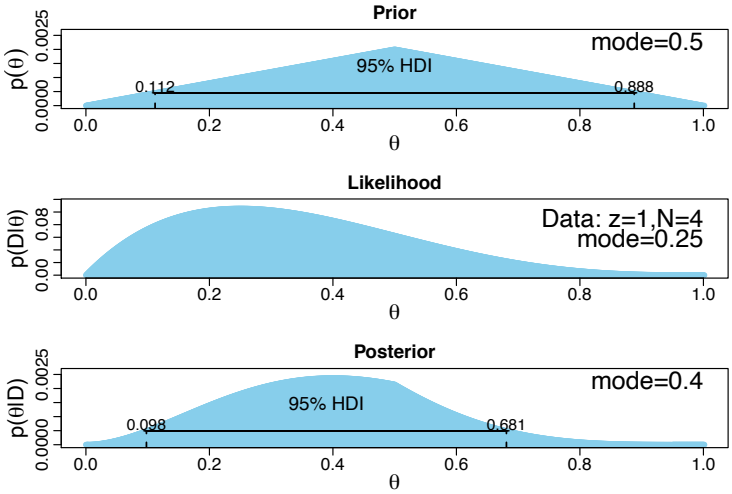
```
Theta = seq( 0 , 1 , length=1001 ) # Fine teeth for Theta.
pTheta = rep(0,length(Theta))      # Only extremes are possible!
pTheta[2] = 1                      # Only extremes are possible!
pTheta[length(pTheta)-1] = 1
pTheta = pTheta/sum(pTheta)        # Make pTheta sum to 1.0
Data = c(rep(0,0),rep(1,1))        # Single flip with 1 head

# openGraph(width=5,height=7)
posterior = BernGrid( Theta, pTheta , Data , plotType="Bars" ,
                      showCentTend="None" , showHDI=FALSE , showpD=FALSE )
```




```
# saveGraph(file="BernGridExample3",type="eps")
```

**Example 4:** this shows the reallocation of the distribution of parameter values. Copmared to the likelihood distribution, the priors distribution has greater impact on the posterior because the sample size is small and the prior is informative .

```
Theta = seq( 0 , 1 , length=1001 )  # Fine teeth for Theta.
pTheta = pmin( Theta , 1-Theta ) # Triangular shape for pTheta.
pTheta = pTheta/sum(pTheta)      # Make pTheta sum to 1.0
Data = c(rep(0,3),rep(1,1))      # 25% heads, N=4

# openGraph(width=5,height=7)
posterior = BernGrid( Theta, pTheta , Data , plotType="Bars" ,
                      showCentTend="Mode" , showHDI=TRUE , showpD=FALSE )
```

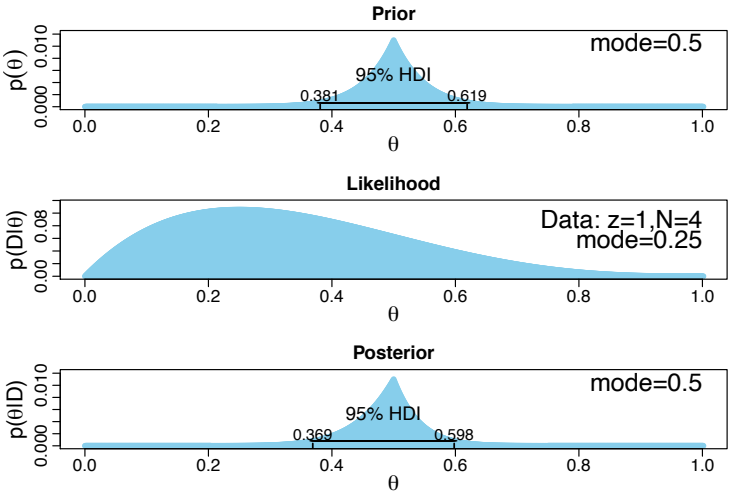



```
# saveGraph(file="BernGridExample4",type="eps")
```

**Example 5:** Similar to the previous example, the prior distribution has greater impact on the posterior because the sample

size is small and the prior is informative. The posterior distribution is similar to the prior distribution

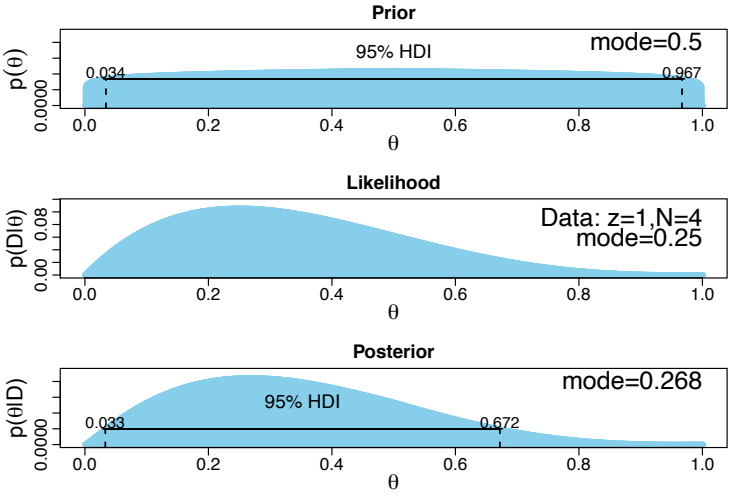
```
Theta = seq( 0 , 1 , length=1001 )  # Fine teeth for Theta.
pTheta = pmin( Theta , 1-Theta ) # Triangular shape for pTheta.
pTheta = pTheta/sum(pTheta)      # Make pTheta sum to 1.0
pTheta = pTheta^10               # Sharpen pTheta !
pTheta = pTheta/sum(pTheta)      # Make pTheta sum to 1.0
Data = c(rep(0,3),rep(1,1))      # 25% heads, N=4

# openGraph(width=5,height=7)
posterior = BernGrid( Theta, pTheta , Data , plotType="Bars" ,
                      showCentTend="Mode" , showHDI=TRUE , showpD=FALSE )
```




```
# saveGraph(file="BernGridExample5",type="eps")
```

**Example 6:** The posterior distribution is similar to the likelihood distribution because the prior distribution is uninformed and vague and the sample size is small.

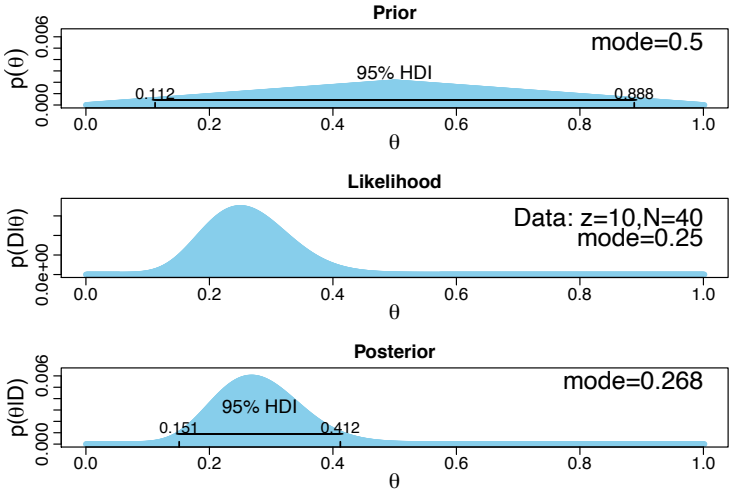
```
Theta = seq( 0 , 1 , length=1001 )  # Fine teeth for Theta.
pTheta = pmin( Theta , 1-Theta ) # Triangular shape for pTheta.
pTheta = pTheta/sum(pTheta)      # Make pTheta sum to 1.0
pTheta = pTheta^0.1              # Flatten pTheta !
pTheta = pTheta/sum(pTheta)      # Make pTheta sum to 1.0
Data = c(rep(0,3),rep(1,1))      # 25% heads, N=4

# openGraph(width=5,height=7)
posterior = BernGrid( Theta, pTheta , Data , plotType="Bars" ,
                      showCentTend="Mode" , showHDI=TRUE , showpD=FALSE )
```



```
# saveGraph(file="BernGridExample6",type="eps")
```

**Example 7:** The posterior distribution is similar to the likelihood distribution because the sample size is big enough to influence the posterior distribution

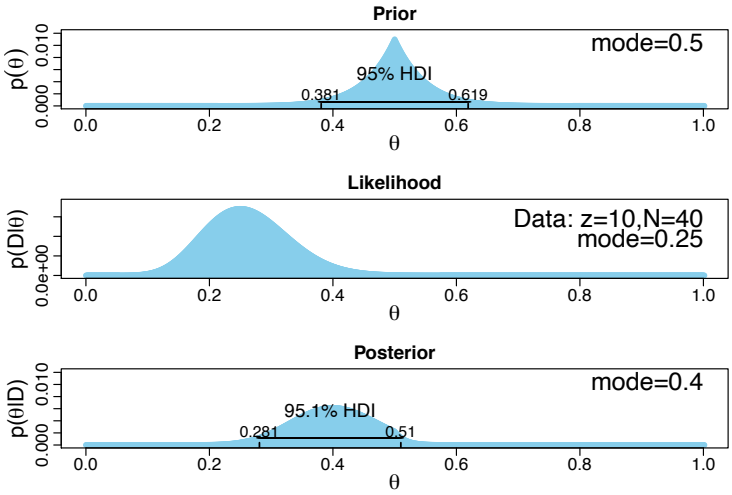
```
Theta = seq( 0 , 1 , length=1001 )  # Fine teeth for Theta.
pTheta = pmin( Theta , 1-Theta ) # Triangular shape for pTheta.
pTheta = pTheta/sum(pTheta)      # Make pTheta sum to 1.0
Data = c(rep(0,30),rep(1,10))    # 25% heads, N=40

# openGraph(width=5,height=7)
posterior = BernGrid( Theta, pTheta , Data , plotType="Bars" ,
                      showCentTend="Mode" , showHDI=TRUE , showpD=FALSE )
```




```
# saveGraph(file="BernGridExample7",type="eps")
```

**Example 8:** The posterior distribution is a compromise between the prior and the likelihood distribution. The prior distribution is informative and the sample size is big enough to influence the posterior distribution

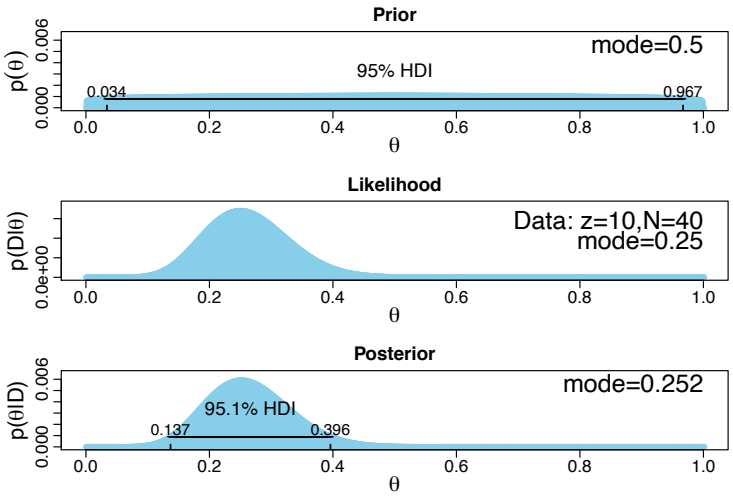
```
Theta = seq( 0 , 1 , length=1001 )  # Fine teeth for Theta.
pTheta = pmin( Theta , 1-Theta ) # Triangular shape for pTheta.
pTheta = pTheta/sum(pTheta)      # Make pTheta sum to 1.0
pTheta = pTheta^10               # Sharpen pTheta !
pTheta = pTheta/sum(pTheta)      # Make pTheta sum to 1.0
Data = c(rep(0,30),rep(1,10))    # 25% heads, N=40
# openGraph(width=5,height=7)
posterior = BernGrid( Theta, pTheta , Data , plotType="Bars" ,
                      showCentTend="Mode" , showHDI=TRUE , showpD=FALSE )
```



```
# saveGraph(file="BernGridExample8",type="eps")
```

**Example 9:** The posterior distribution is similar to the likelihood distribution because the prior distribution is uninformed (too flat) and the sample size is big enough to influence the posterior distribution

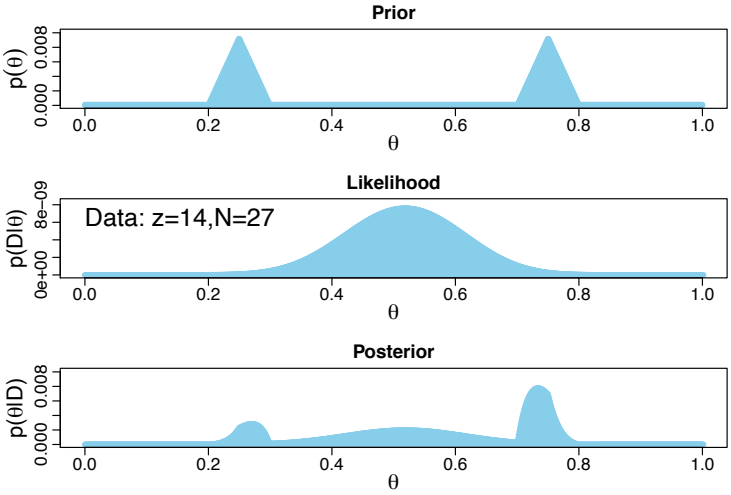
```
Theta = seq( 0 , 1 , length=1001 )  # Fine teeth for Theta.
pTheta = pmin( Theta , 1-Theta ) # Triangular shape for pTheta.
pTheta = pTheta/sum(pTheta)      # Make pTheta sum to 1.0
pTheta = pTheta^0.1              # Flatten pTheta !
pTheta = pTheta/sum(pTheta)      # Make pTheta sum to 1.0
Data = c(rep(0,30),rep(1,10))    # 25% heads, N=40

# openGraph(width=5,height=7)
posterior = BernGrid( Theta, pTheta , Data , plotType="Bars" ,
                      showCentTend="Mode" , showHDI=TRUE , showpD=FALSE )
```




```
# saveGraph(file="BernGridExample9",type="eps")
```

**Example 10:** Similar to the previous example, the posterior distribution is a compromise between the prior and the likelihood distribution. The prior distribution is informative and the sample size is big enough to influence the posterior distribution.

```
Theta = seq( 0 , 1 , length=1000 )  # Fine teeth for Theta.
# Two triangular peaks on a small non-zero floor:
pTheta = c( rep(1,200),seq(1,100,length=50),seq(100,1,length=50),rep(1,200) ,
            rep(1,200),seq(1,100,length=50),seq(100,1,length=50),rep(1,200) )
pTheta = pTheta/sum(pTheta)      # Make pTheta sum to 1.0
Data = c(rep(0,13),rep(1,14))

# openGraph(width=5,height=7)
posterior = BernGrid( Theta, pTheta , Data , plotType="Bars" ,
                      showCentTend="None" , showHDI=FALSE , showpD=FALSE )
```




```
# saveGraph(file="BernGridExample10",type="eps")
```

## Class Exercise

### what happens when you have 3000 heads out of 12000 flips

the BernGrid function won't work because the following code in the function won't work.

```
# Create summary values of Data
z = sum( Data==1 ) # number of 1's in Data
N = length( Data ) # number of flips in Data
# Compute the likelihood of the Data for each value of Theta.
pDataGivenTheta = Theta^z * (1-Theta)^(N-z)
# Compute the evidence and the posterior.
pData = sum( pDataGivenTheta * pTheta )
pThetaGivenData = pDataGivenTheta * pTheta / pData
```

When you have the following parameters, Theta^z and (1-Theta)^(N-z) in the code above will be infitely small (zero). Consequently, pDataGivenTheta and pData will be zero and pThetaGivenData will be undefined since the denominator will be zero.

```
Theta = seq(0,1,by = .1)
pTheta = pmin(Theta, 1-Theta)
pTheta = pTheta/sum(pTheta)
Data = c(rep(0,12000-3000),rep(1,3000)) # 30 heads, out of 120 flips
# BernGrid(Theta,pTheta, Data = Data)
```