

# bayes-hw-wk2-kenwan

## Wk2 - assignment. Posterior distribution for binomial.

Repeat calculations of Section 5 of the workshop:

Find posterior probability for binomial model with uniform prior and data. Use `set.seed(81)` for simulation of  $\theta$

Set likelihood function for binom.

```
likeli<-function(par,data){  
  sdata<-sum(data)  
  ldata<-length(data)  
  return(par^sdata*(1-par)^(ldata-sdata))  
}
```

Generate theta. Uniform - 1000 samples.

```
set.seed(81)  
(data1<-rbinom(10,1,.71))
```

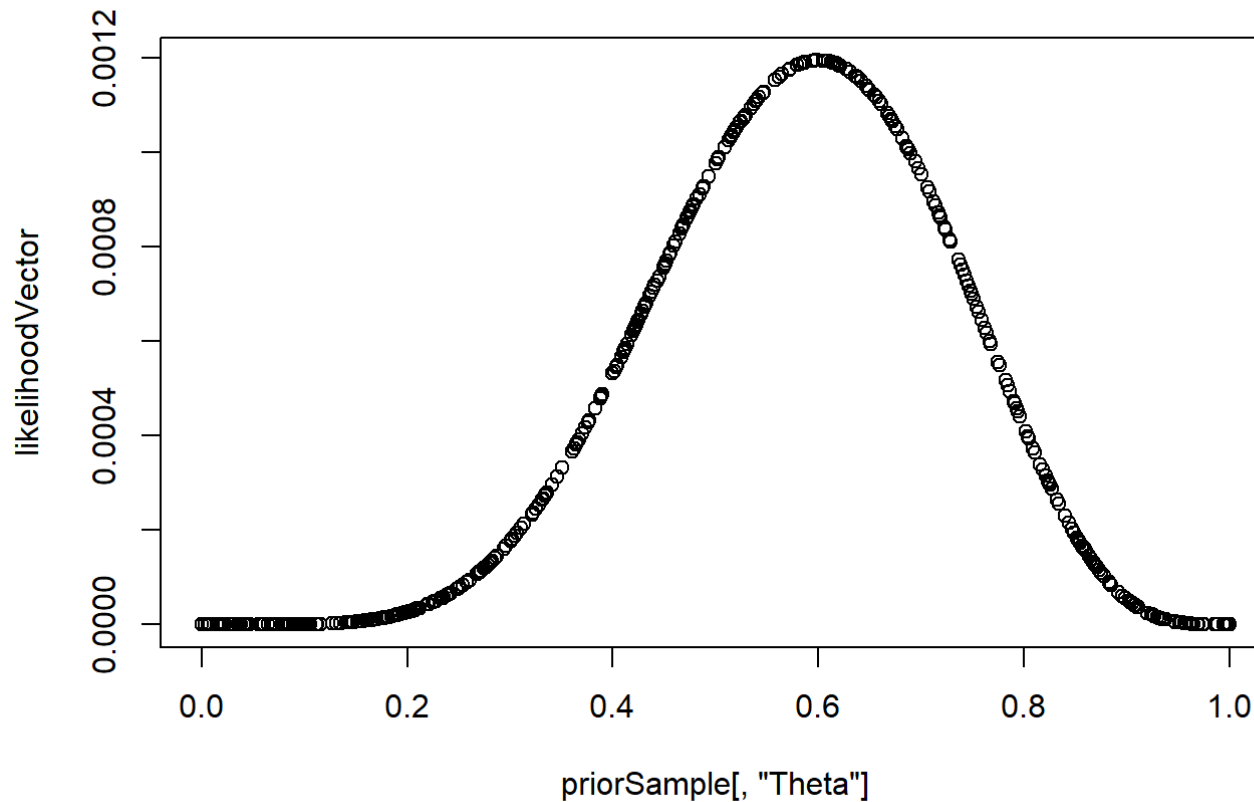
```
## [1] 1 0 1 1 0 0 0 1 1 1
```

```
Theta = seq( .00001 , 1 -.00001, 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
```

Now data generation and sample of theta from prior.

```
priorInd<-sample(1:length(Theta),500,replace = T)  
priorSample<-cbind(Theta=Theta[priorInd],Prob=pTheta[priorInd])  
priorSample<-rbind(priorSample,  
  c(head(Theta,1),head(pTheta,1)),
```

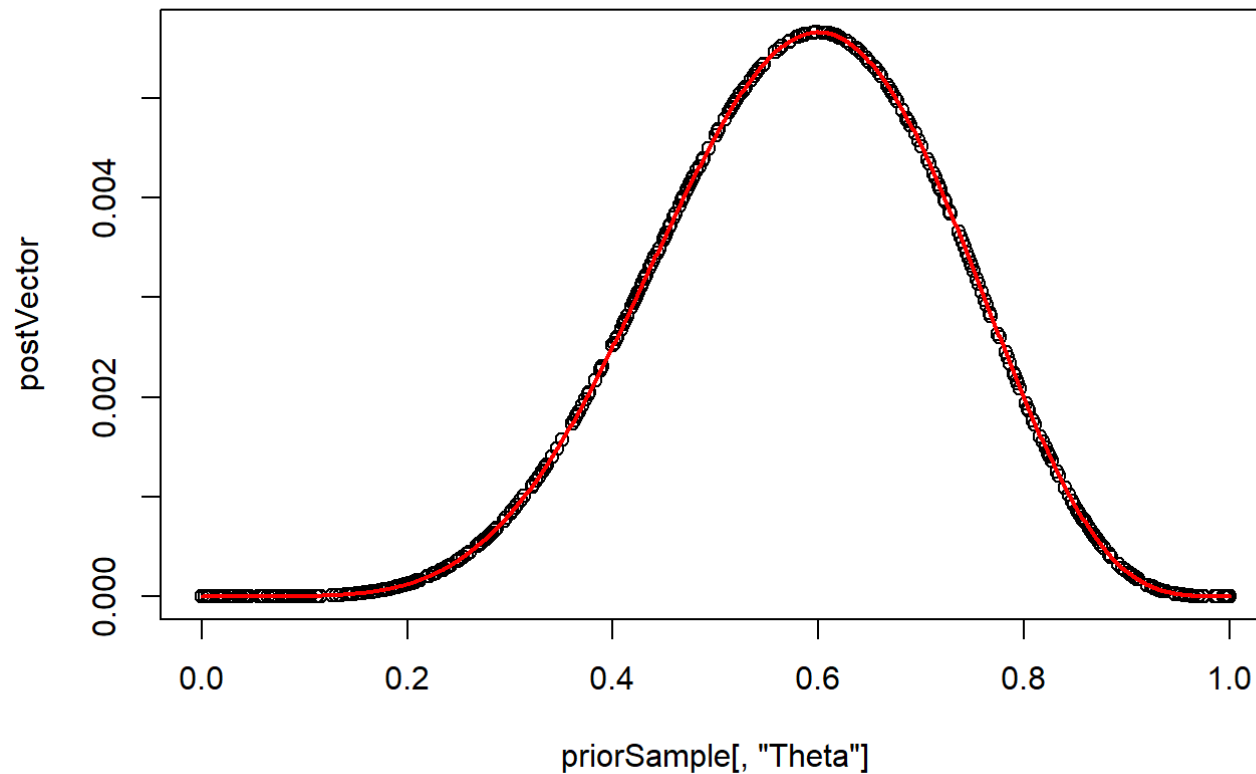
```
c(tail(Theta,1),tail(pTheta,1)))  
  
likelihoodVector<-sapply(priorSample[, "Theta"], function(z) likeli(z,data1))  
plot(priorSample[, "Theta"],likelihoodVector)
```



Now posterior.

```
postVector<-priorSample[, "Prob"]*likelihoodVector  
postVector<-postVector/sum(postVector)  
plot(priorSample[, "Theta"],postVector)
```

```
postDistr<-approxfun(priorSample[, "Theta"], postVector, method="linear")  
plot(priorSample[, "Theta"], postVector)  
lines(Theta, postDistr(Theta), col="red", lwd=2)
```



Now we have our posterior sample as well.

Mode mean var

```
postDistr1<-postDistr(Theta)
(model<-Theta[which.max(postDistr(Theta))])
```

```
## [1] 0.597998
```

```
(mean1<-Theta*%postDistr(Theta)/sum(postDistr(Theta)))
```

```
##           [,1]
## [1,] 0.5833332
```

```
(var1<-((Theta-mean1)^2)*%postDistr(Theta)/sum(postDistr(Theta)))
```

```
## Warning in Theta - mean1: Recycling array of length 1 in vector-array arithmetic is deprecated.
## Use c() or as.vector() instead.
```

```
##           [,1]
## [1,] 0.01870004
```

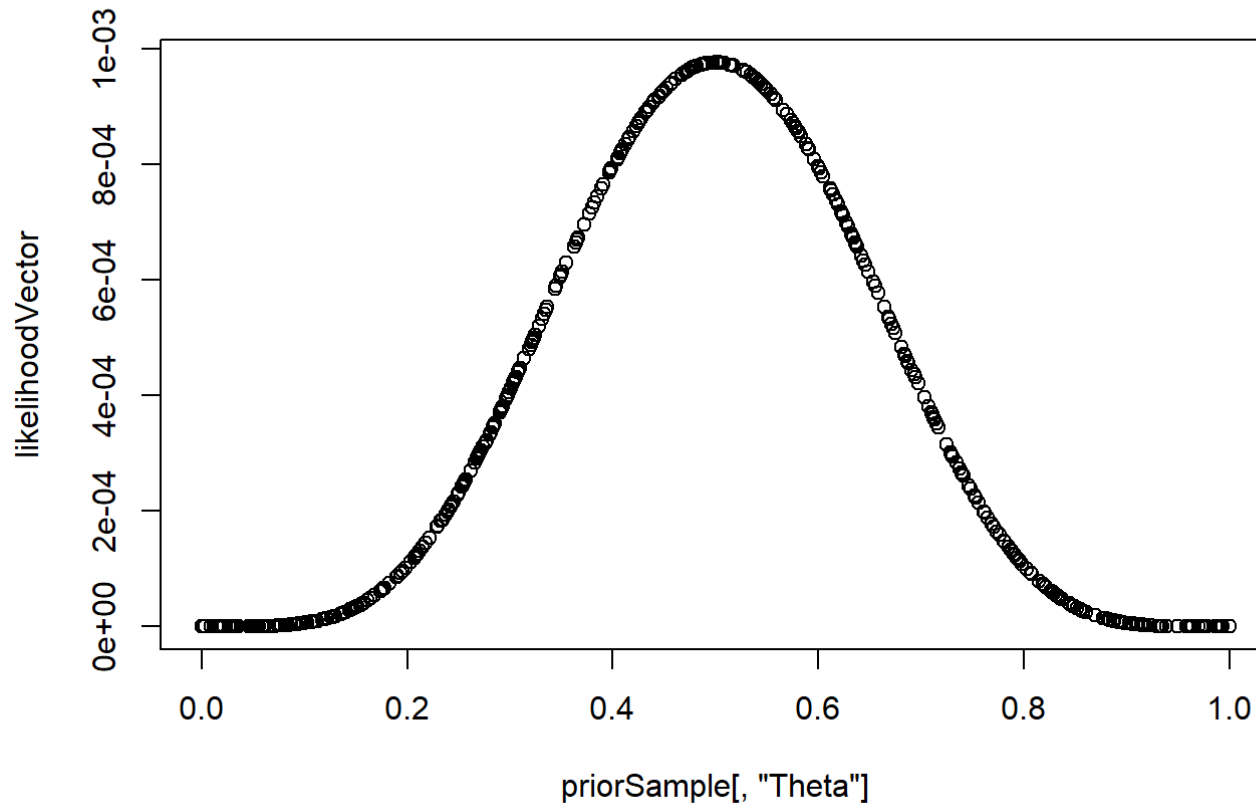
## Now data generation part 2.

```
set.seed(97)
pTheta<-postDistr(Theta)/sum(postDistr(Theta))
# new data generation
(data2<-rbinom(10,1,.71))
```

```
## [1] 1 1 0 1 0 0 0 0 1 1
```

```
priorInd<-sample(1:length(Theta),500,replace = T)
priorSample<-cbind(Theta=Theta[priorInd],Prob=pTheta[priorInd])
```

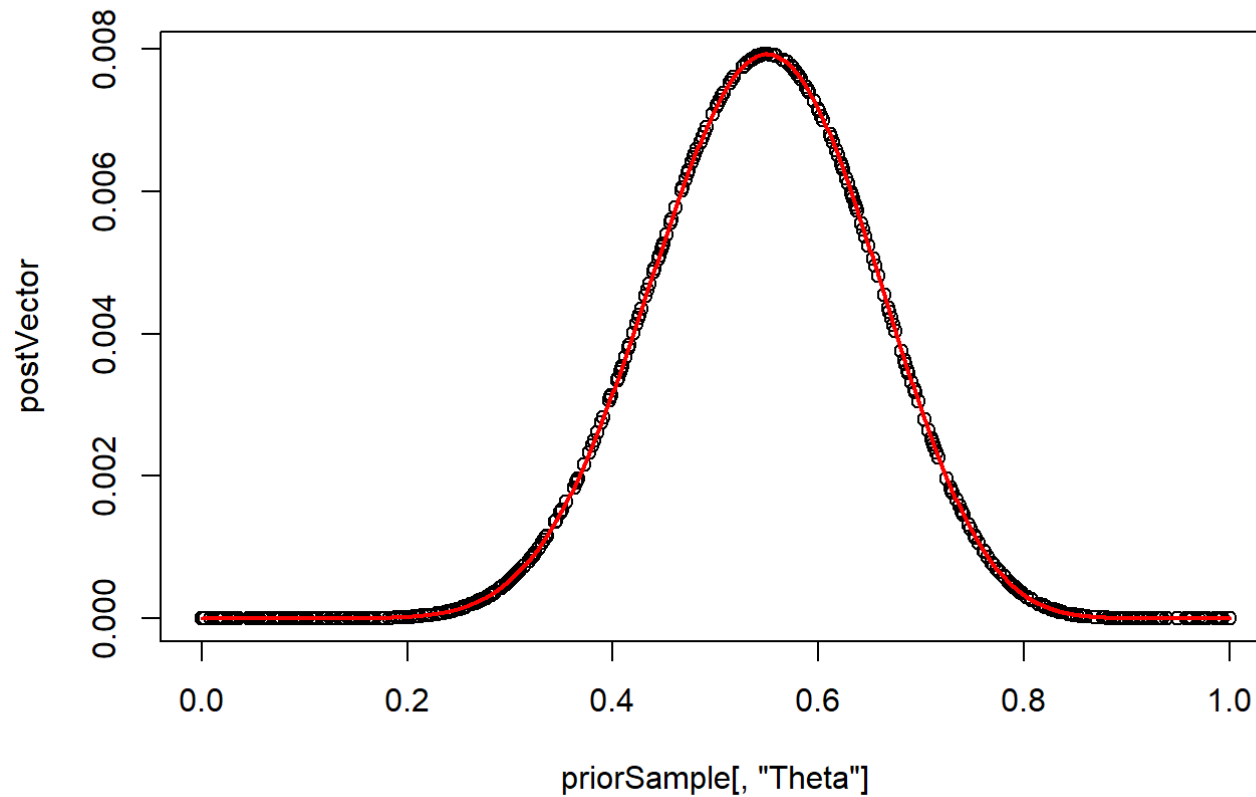
```
priorSample<-rbind(priorSample,  
                  c(head(Theta,1),head(pTheta,1)),  
                  c(tail(Theta,1),tail(pTheta,1)))  
  
likelihoodVector<-sapply(priorSample[, "Theta"], function(z) likeli(z,data2))  
plot(priorSample[, "Theta"], likelihoodVector)
```



New posterior.

```
postVector<-priorSample[, "Prob"]*likelihoodVector
postVector<-postVector/sum(postVector)
plot(priorSample[, "Theta"], postVector)

postDistr<-approxfun(priorSample[, "Theta"], postVector, method="linear")
plot(priorSample[, "Theta"], postVector)
lines(Theta, postDistr(Theta), col="red", lwd=2)
```



```
postDistr2<-postDistr(Theta)
(mode2<-Theta[which.max(postDistr(Theta))])
```

```
## [1] 0.547999
```

```
(mean2<-Theta%%postDistr(Theta)/sum(postDistr(Theta)))
```

```
##           [,1]
## [1,] 0.5454511
```

```
(var2<-((Theta-mean2)^2)%*%postDistr(Theta)/sum(postDistr(Theta)))
```

```
## Warning in Theta - mean2: Recycling array of length 1 in vector-array arithmetic is deprecated.
## Use c() or as.vector() instead.
```

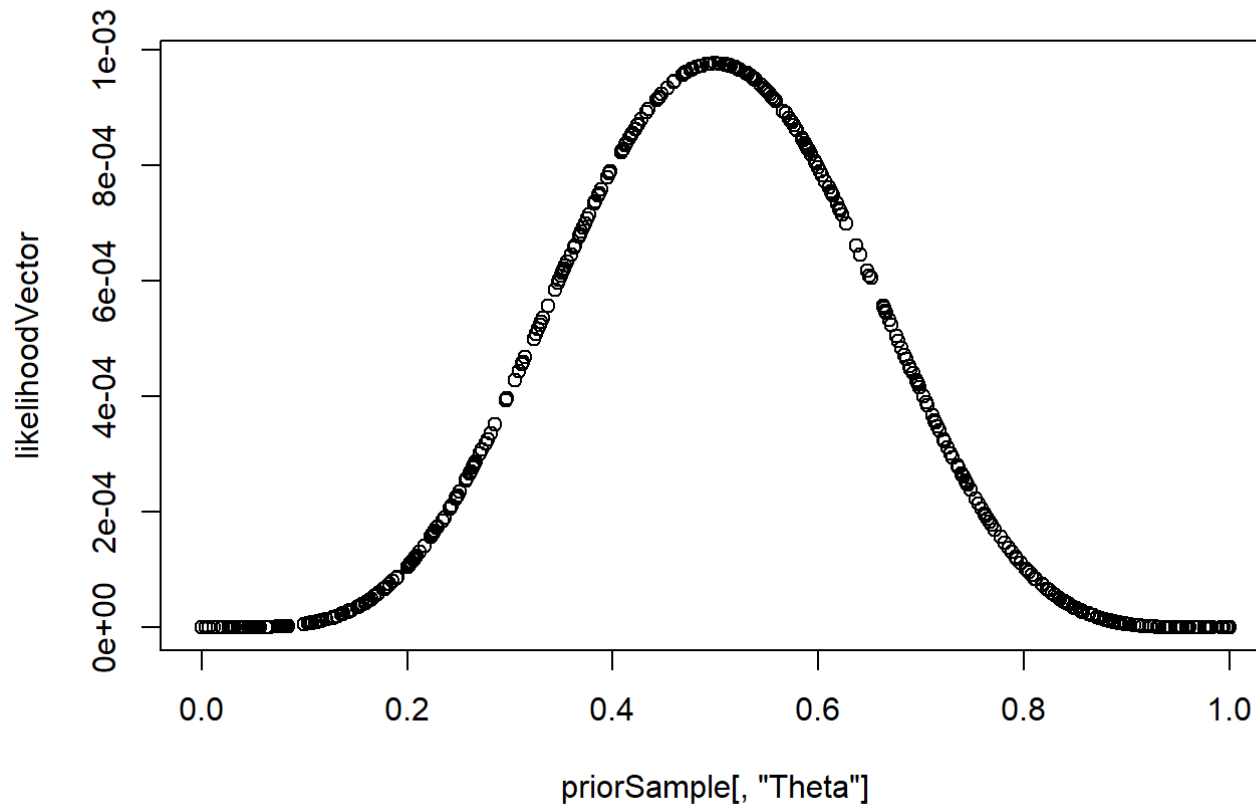
```
##           [,1]
## [1,] 0.01078324
```

## Now repeat for the reverse.

### Now data generation and sample of theta from prior using data2

```
priorInd<-sample(1:length(Theta),500,replace = T)
priorSample<-cbind(Theta=Theta[priorInd],Prob=pTheta[priorInd])
priorSample<-rbind(priorSample,
                   c(head(Theta,1),head(pTheta,1)),
                   c(tail(Theta,1),tail(pTheta,1)))

likelihoodVector<-sapply(priorSample[, "Theta"],function(z) likeli(z,data2))
plot(priorSample[, "Theta"],likelihoodVector)
```

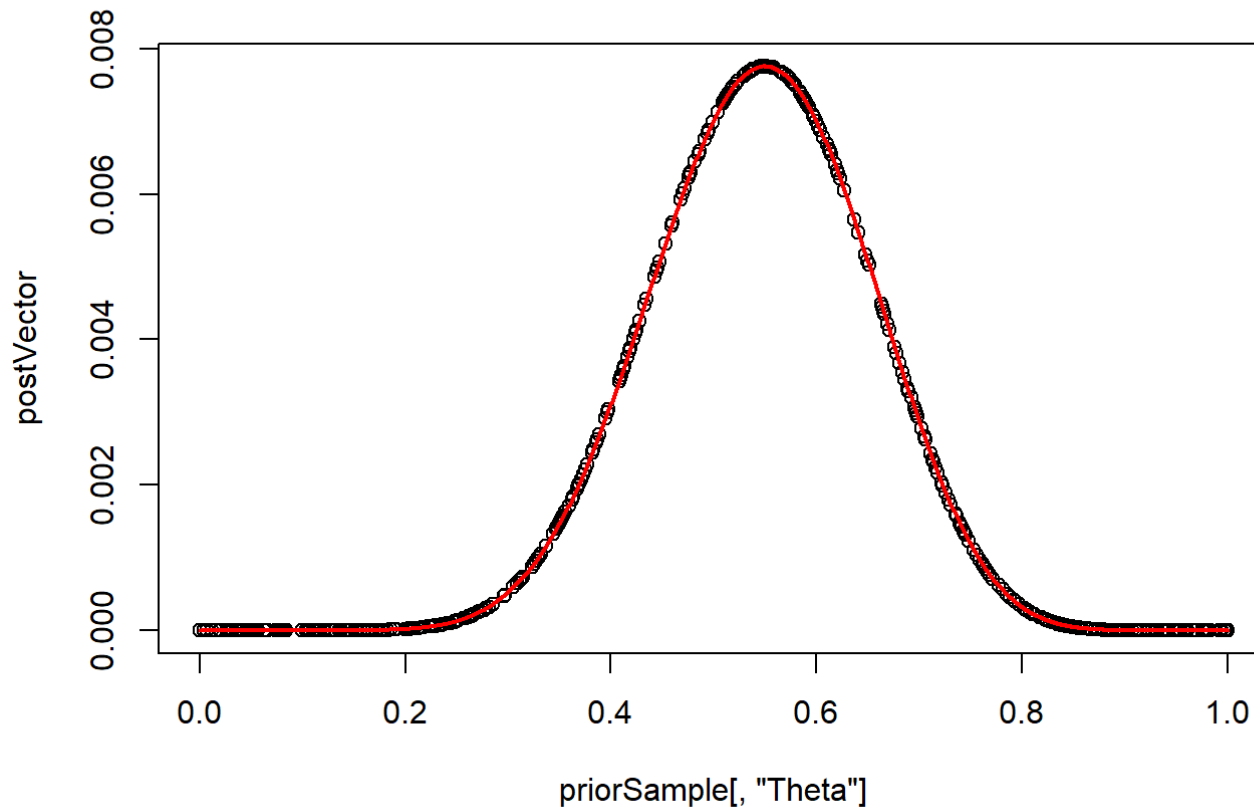


Now posterior.

```
postVector<-priorSample[, "Prob"]*likelihoodVector
postVector<-postVector/sum(postVector)
plot(priorSample[, "Theta"], postVector)

postDistr<-approxfun(priorSample[, "Theta"], postVector, method="linear")
plot(priorSample[, "Theta"], postVector)
lines(Theta, postDistr(Theta), col="red", lwd=2)
```





Mode mean var

```
postDistr3<-postDistr(Theta)  
(mode3<-Theta[which.max(postDistr(Theta))])
```

```
## [1] 0.547999
```

```
(mean3<-Theta%*%postDistr(Theta)/sum(postDistr(Theta)))
```

```
##           [,1]
## [1,] 0.5454436
```

```
(var3<-((Theta-mean3)^2)%*%postDistr(Theta)/sum(postDistr(Theta)))
```

```
## Warning in Theta - mean3: Recycling array of length 1 in vector-array arithmetic is deprecated.
## Use c() or as.vector() instead.
```

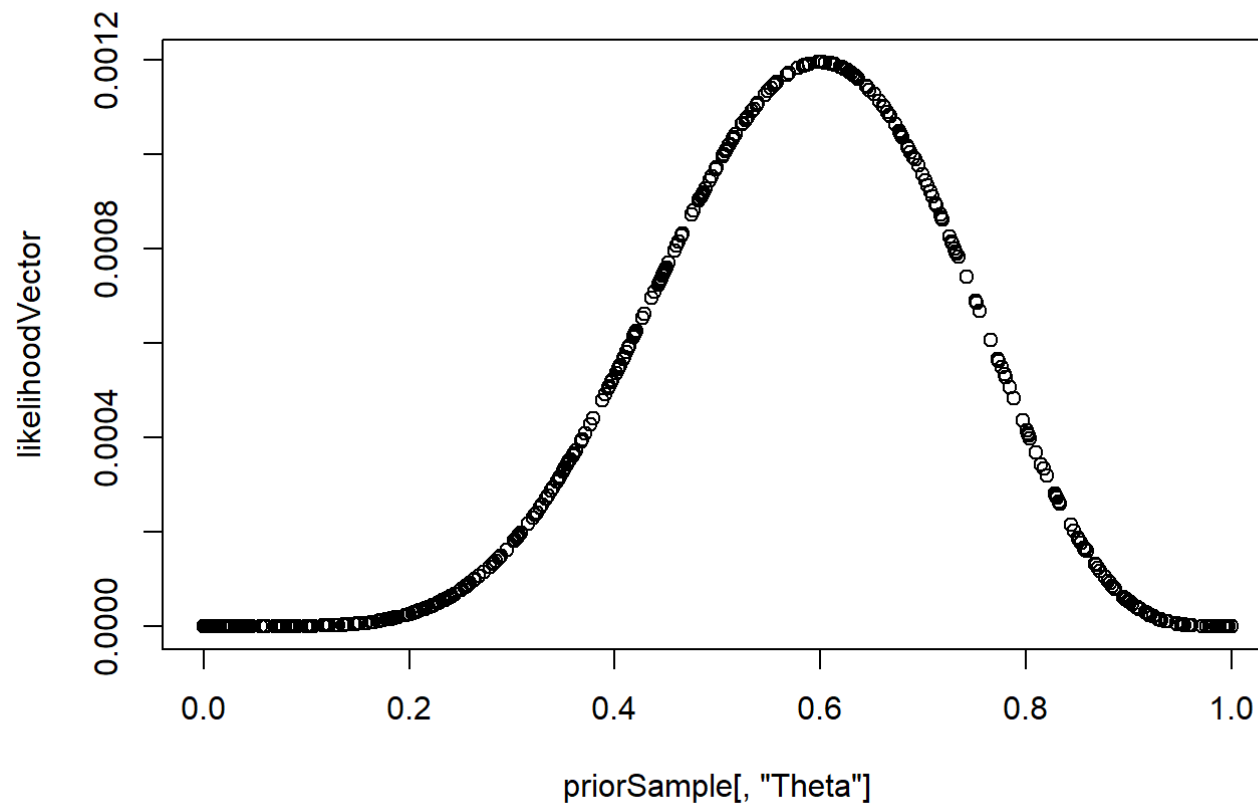
```
##           [,1]
## [1,] 0.01078349
```

## Reupdate with data1 now

```
pTheta<-postDistr(Theta)/sum(postDistr(Theta))

priorInd<-sample(1:length(Theta),500,replace = T)
priorSample<-cbind(Theta=Theta[priorInd],Prob=pTheta[priorInd])
priorSample<-rbind(priorSample,
                   c(head(Theta,1),head(pTheta,1)),
                   c(tail(Theta,1),tail(pTheta,1)))

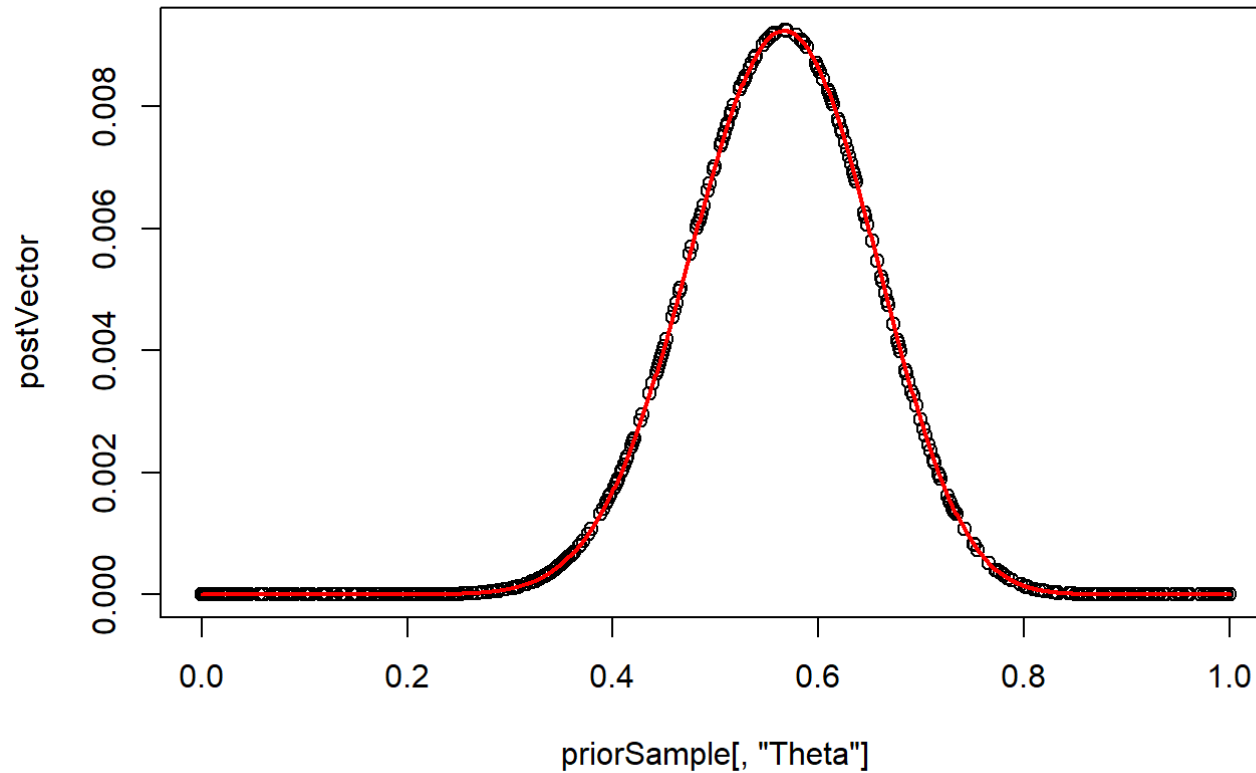
likelihoodVector<-sapply(priorSample[, "Theta"],function(z) likeli(z,data1))
plot(priorSample[, "Theta"],likelihoodVector)
```



New posterior.

```
postVector<-priorSample[, "Prob"]*likelihoodVector
postVector<-postVector/sum(postVector)
plot(priorSample[, "Theta"], postVector)

postDistr<-approxfun(priorSample[, "Theta"], postVector, method="linear")
plot(priorSample[, "Theta"], postVector)
lines(Theta, postDistr(Theta), col="red", lwd=2)
```



```
postDistr4<-postDistr(Theta)  
(mode4<-Theta[which.max(postDistr(Theta))])
```

```
## [1] 0.5669987
```

```
(mean4<-Theta*%postDistr(Theta)/sum(postDistr(Theta)))
```

```
##           [,1]  
## [1,] 0.5625066
```

```
(var4<-((Theta-mean4)^2)*%postDistr(Theta)/sum(postDistr(Theta)))
```

```
## Warning in Theta - mean4: Recycling array of length 1 in vector-array arithmetic is deprecated.  
## Use c() or as.vector() instead.
```

```
##           [,1]  
## [1,] 0.007463951
```

## Comparison

```
final_comp = rbind(data.frame(mode = mode2,mean = mean2,var = var2),data.frame(mode=mode4,mean = mean4, var = var  
4))
```