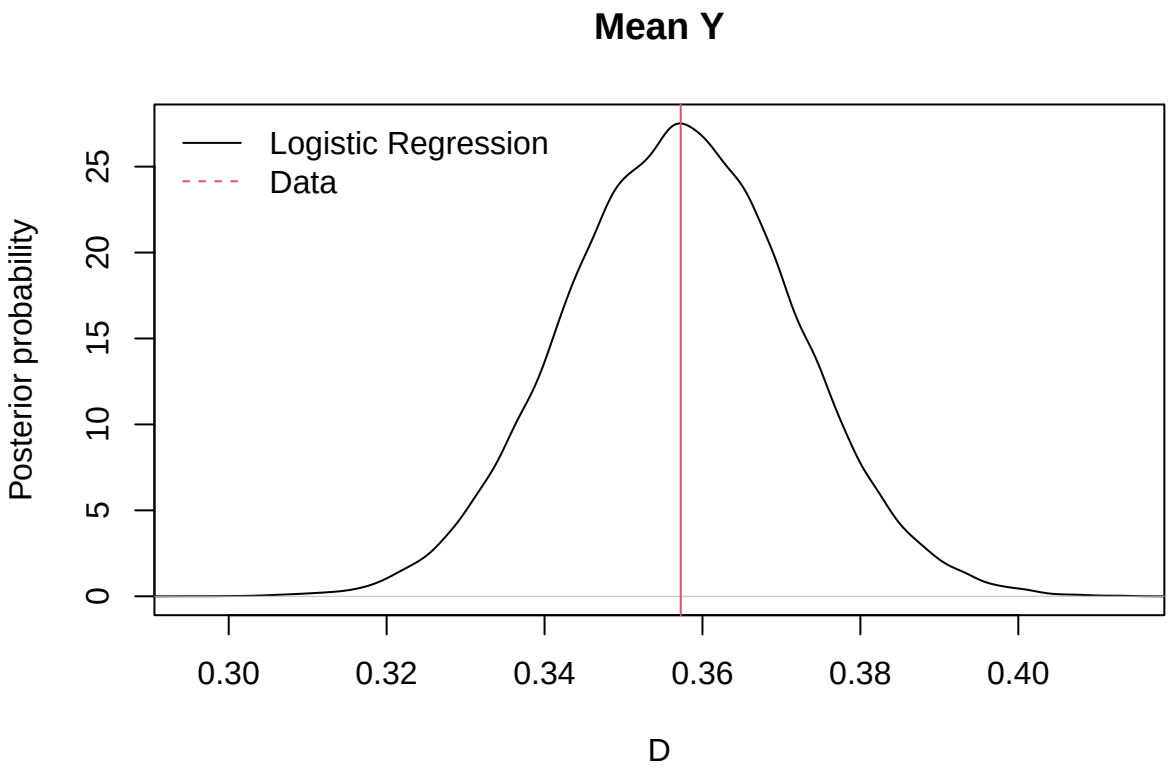
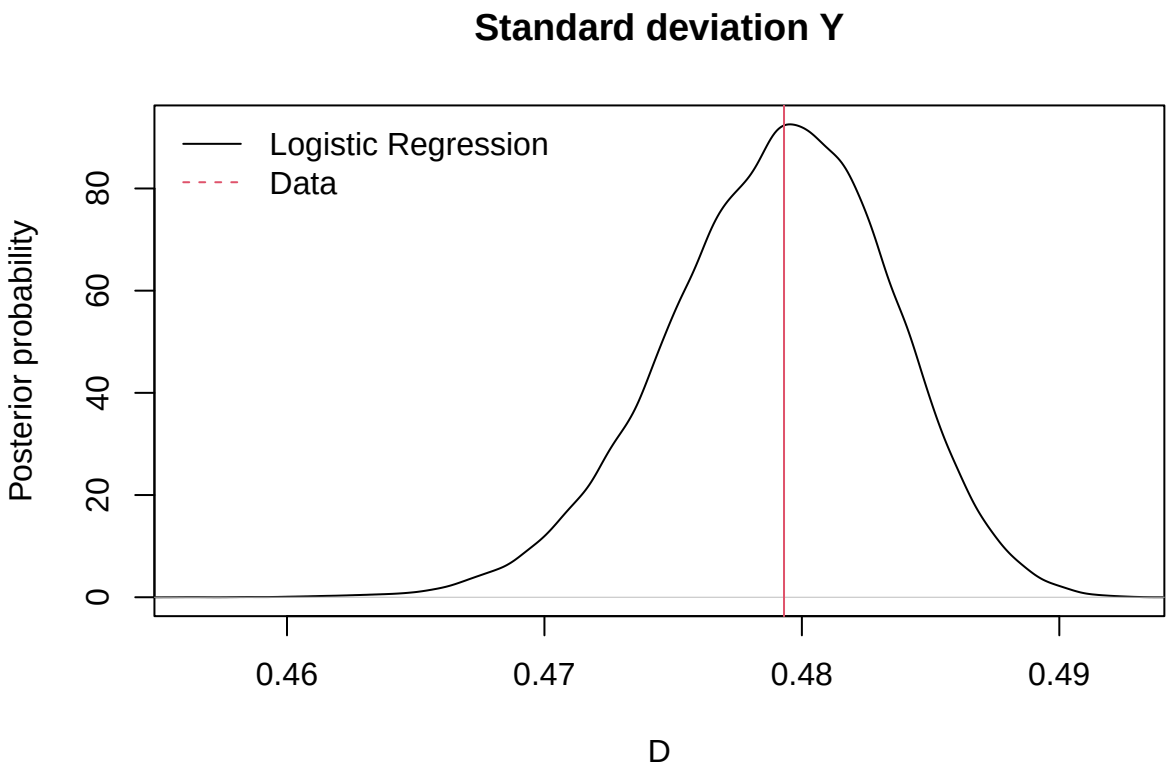
Problem 1 Fit logistic regression model to the `gambia` data in the previous question and use posterior predictive checks to verify the model fits well. If you find model misspecification, suggest (but do not fit) alternative.

There are many possible options to choose. One of them is to test the mean.

```
data(gambia)
Y = gambia[,3]
X = scale(gambia[,4:8])
s = gambia[,1:2]
n = length(Y)
p = ncol(X)
# Compute the village ID
S = unique(s) # Lat/long of the villages
m = nrow(S)
village = rep(0,n)
members = rep(0,m)
for(j in 1:m){
d = (s[,1]-S[j,1])^2 + (s[,2]-S[j,2])^2
village[d==0] = j
members[j] = sum(d==0)
}

set.seed(0)
burn = 10000
n.iter = 20000
n.chains = 2
params = c("D","beta")

# Fit logistic model
model_string = textConnection("model{
for(i in 1:n){
Y[i] ~ dbern(pi[i])
logit(pi[i]) = beta[1] + X[i,1]*beta[2] +
X[i,2]*beta[3] + X[i,3]*beta[4] +
X[i,4]*beta[5] + X[i,5]*beta[6]
}
for(j in 1:6){beta[j] ~ dnorm(0,0.01)}

#Posterior predictive checks
for(i in 1:n){
Y2[i] ~ dbern(pi[i])
}
D[1] = mean(Y2[])
D[2] =sd(Y2[])


}")
data = list(Y=Y,X=X,n=length(Y))
model = jags.model(model_string,data = data, n.chains=n.chains,quiet=TRUE)
update(model, burn, progress.bar="none")
samples = coda.samples(model, variable.names=params,n.iter=n.iter, progress.bar="none")
```

```r
D1  = rbind(samples[[1]], samples[[2]])
D0   = c(mean(Y),   sd(Y))
Dnames = c("Mean Y", "Standard deviation Y")

# Compute the test stats for the models

pval1 = rep(0,2)
names(pval1)=Dnames

for(j in 1:2){
  plot(density(D1[,j]),xlim=range(c(D0[j],D1[,j])),
                       xlab="D",ylab="Posterior probability",
                       main=Dnames[j])
  abline(v=D0[j],col=2)
  legend("topleft",c("Logistic Regression","Data"),lty=1:2,col=1:2,bty="n")

  pval1[j] = mean(D1[,j]>D0[j])
}
```






```
pval1
```

```
##               Mean Y Standard deviation Y
##             0.490075              0.490075
```

From plots we observe that data statistics is middle of the distribution in both test statistics. Moreover, the Bayesian p-values are close to 0.5 so we conclude that logistic regression model fits to the given data well.

Problem 2

Fit model

$$\mathcal{M}_{\in} : \text{logit}(p_i) = \beta_1 + \beta_2 \text{logit}(q_i)$$

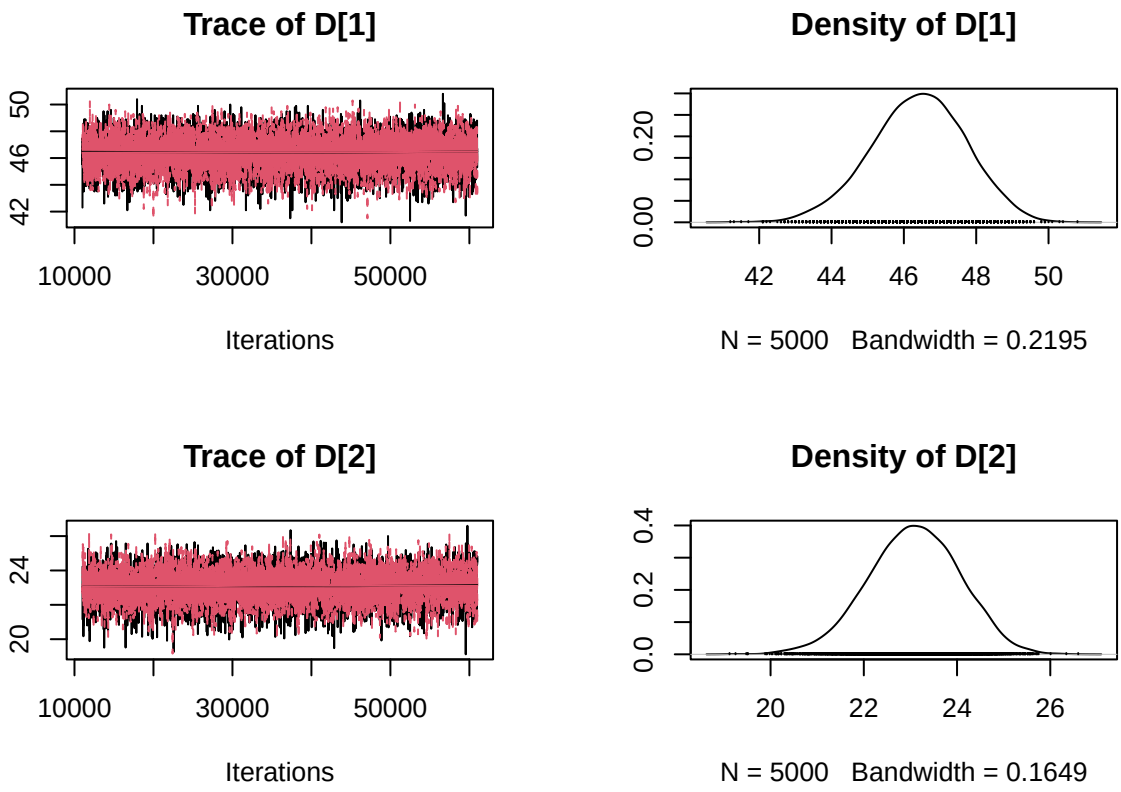
to the NBA data in the previous question and use posterior predictive checks to verify the model fits well. If you find model misspecification, suggest (but do not fit) alternatives.

Solution

There might be multiple solutions for this problem. This is my approach for this problem.

```
# Load the data
 Y      <- c(64,72,55,27,75,24,28,66,40,13)
 n      <- c(75,95,63,39,83,26,41,82,54,16)
 q      <- c(0.845,0.847,0.880,0.674,0.909,
             0.898,0.770,0.801,0.802,0.875)
 X      <- log(q/(1-q))
```

```
# Define the SSVS model:
 nba_model <- "model{
  for(i in 1:10){
    Y[i]          ~ dbinom(pi[i],n[i])
    logit(pi[i]) <- beta1 + beta2*X[i]
  }
  beta1 ~ dnorm(0,0.1)
  beta2 = 1+delta*gamma
  gamma ~ dbern(0.5)
  delta ~ dnorm(0,tau)
  tau   ~ dgamma(0.1,0.1)
#Posterior predictive checks
for(i in 1:10){
Y2[i] ~ dbinom(pi[i],n[i])
}
D[1] <- mean(Y2[])
D[2] <- sd(Y2[])}"




 mod <- textConnection(nba_model)
 data  <- list(Y=Y,X=X,n=n)
 model <- jags.model(mod,data = data, n.chains=2,quiet=TRUE)
 update(model, 10000, progress.bar="none")
 samps <- coda.samples(model, variable.names=c("D"),
                       n.iter=50000, thin=10,progress.bar="none")
 plot(samps)
```

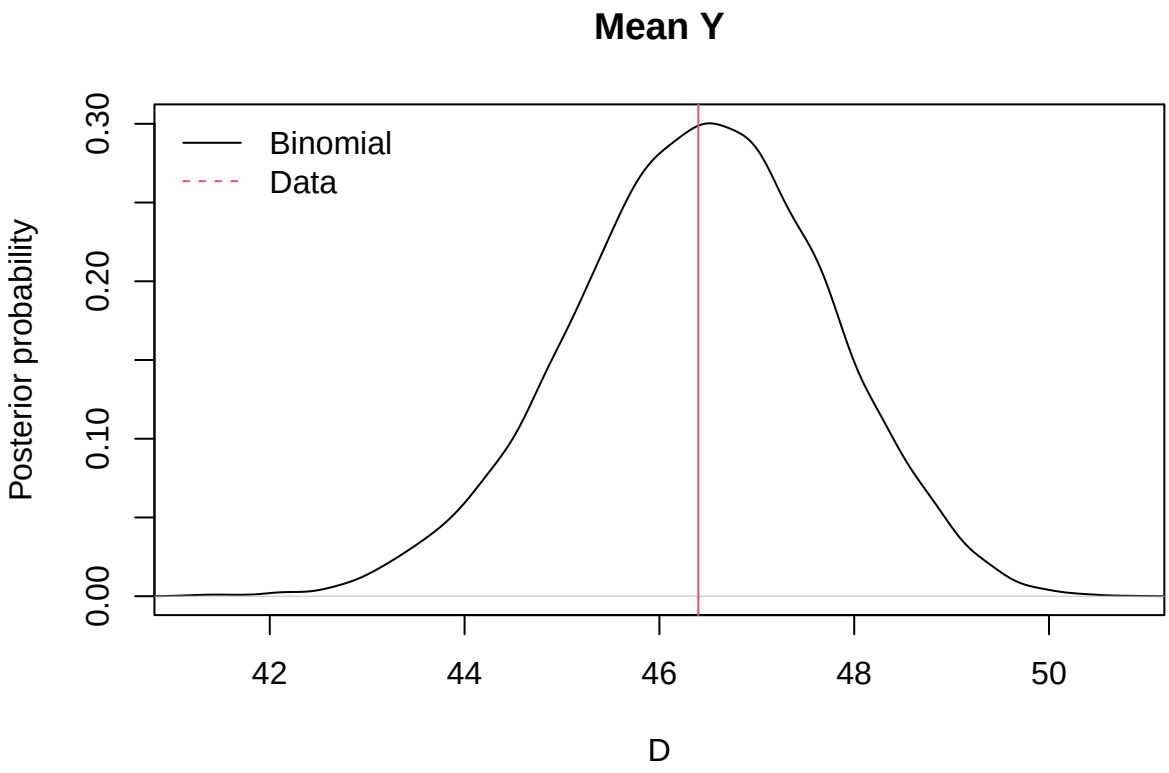


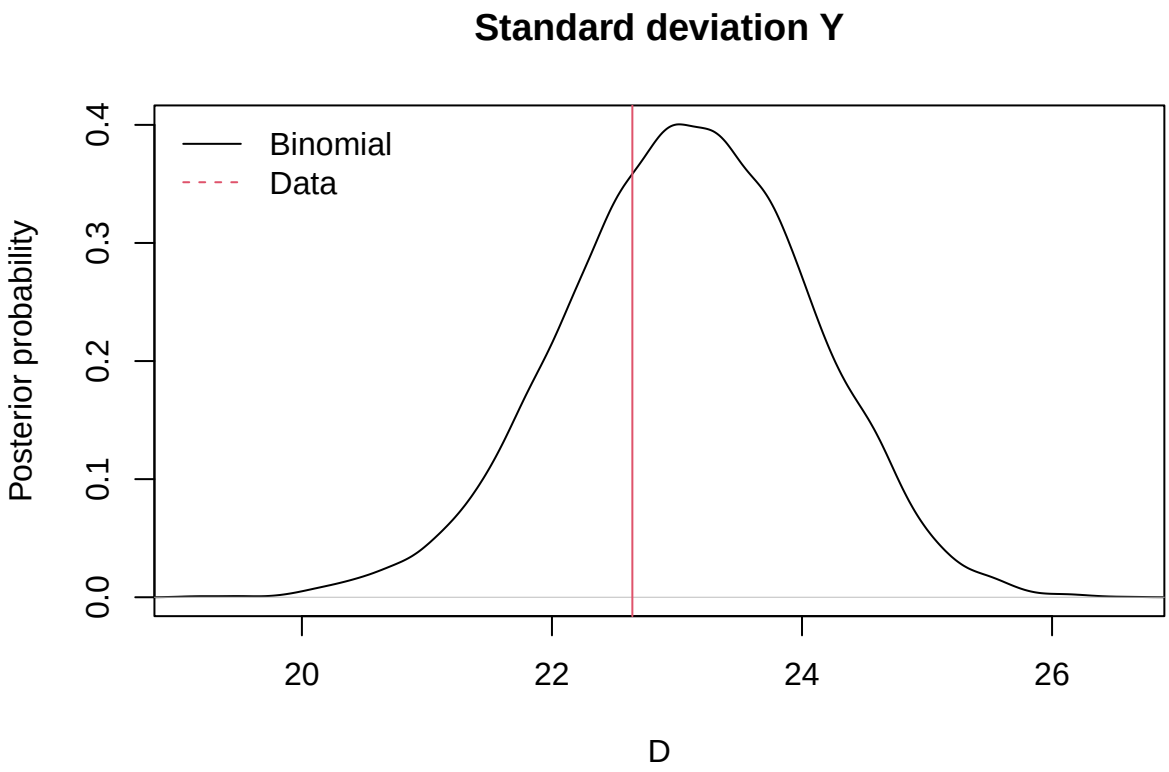

```
print(summary(samps))
```

```
##
## Iterations = 11010:61000
## Thinning interval = 10
## Number of chains = 2
## Sample size per chain = 5000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##       Mean     SD Naive SE Time-series SE
## D[1] 46.40 1.3065 0.013065       0.013066
## D[2] 23.08 0.9813 0.009813       0.009812
##
## 2. Quantiles for each variable:
##
##       2.5%   25%   50%   75% 97.5%
## D[1] 43.70 45.50 46.40 47.30 48.90
## D[2] 21.09 22.43 23.09 23.75 24.92
```

```
D1  <- rbind(samps[[1]], samps[[2]])

D0   <- c(mean(Y),   sd(Y))
Dnames <- c("Mean Y", "Standard deviation Y")
```

```r
# Compute the test stats for the models

pval1 <- rep(0,2)
names(pval1)<-Dnames

for(j in 1:2){
  plot(density(D1[,j]),xlim=range(c(D0[j],D1[,j])),
                       xlab="D",ylab="Posterior probability",
                       main=Dnames[j])
 abline(v=D0[j],col=2)
 legend("topleft",c("Binomial","Data"),lty=1:2,col=1:2,bty="n")

 pval1[j] <- mean(D1[,j]>D0[j])
}
```






```
pval1
```

```
##              Mean Y Standard deviation Y
##              0.4982               0.6759
```

From plots we observe that data statistics is middle of the distribution in both test statistics. Moreover, the Bayesian $p-$values are close to 0.5 so we conclude that binomial model fits to the given data well.