Posterior predictive checks

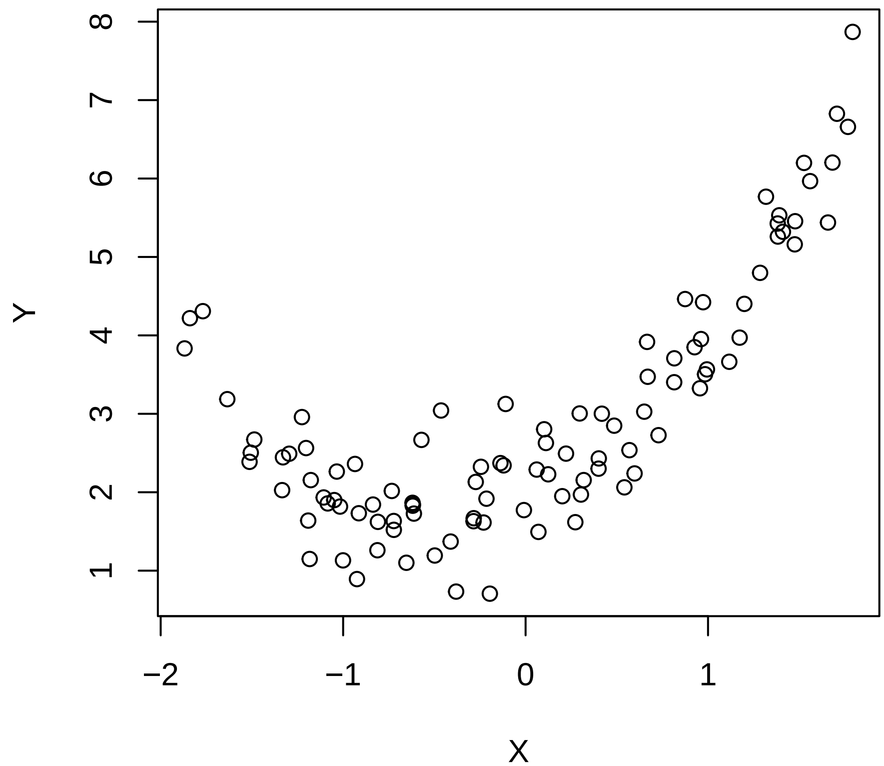
Arnab Hazra

4/9/2023

Here we simulate 100 observations from a quadratic regression $Y_i = \mu_i + \epsilon_i, i = 1, \ldots, 100$. Here, $\mu_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2$ and $\epsilon_i \sim \text{Normal}(0, \sigma^2)$.

1. Simulate data

```
rm(list = ls())

n <- 100

set.seed(100)

X <- runif(n)
X <- as.vector(scale(X))

beta <- c(2, 1, 1) # (beta0, beta1, beta2)
sigmaSq <- 0.25

Y <- beta[1] + beta[2] * X + beta[3] * X^2 + rnorm(n, mean = 0, sd = sqrt(sigmaSq))

plot(X, Y)
```



Now, we want to compare the models $\mathcal{M}_1 : \mu_i = \beta_0 + \beta_1 X_i$ versus $\mathcal{M}_2 : \mu_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2$.

Posterior predictive checks are performed for the following test statistics:

$D_1(Y) = \min(Y_1, \dots, Y_n)$ $D_2(Y) = mean(Y_1, \dots, Y_n)$ $D_3(Y) = \max(Y_1, \dots, Y_n)$

2. Specify two competing models:

Model 1 has non-informative Gaussian priors $\beta_0, \beta_1 \sim \text{Normal}(0, 100^2)$.

Model 2 has non-informative Gaussian priors $\beta_0, \beta_1, \beta_2 \sim \text{Normal}(0, 100^2)$.

```
library(rjags)
```

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

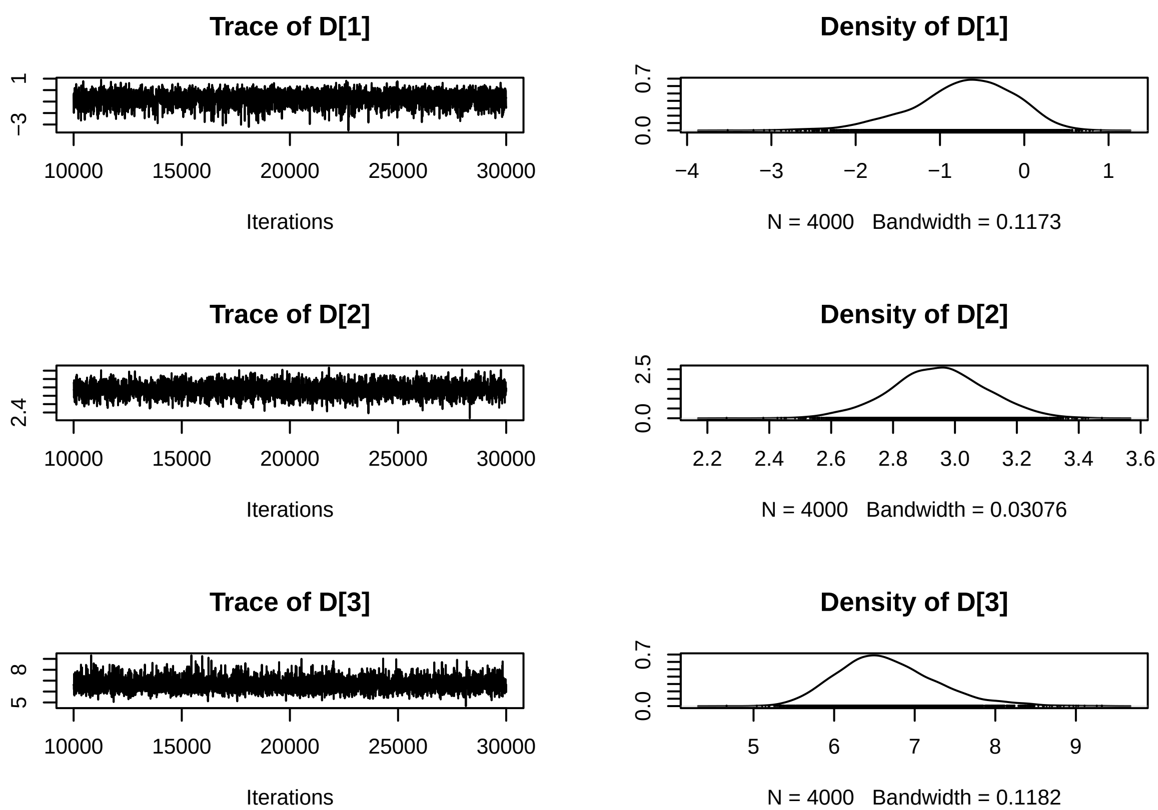
```
## Linked to JAGS 4.3.0
```

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

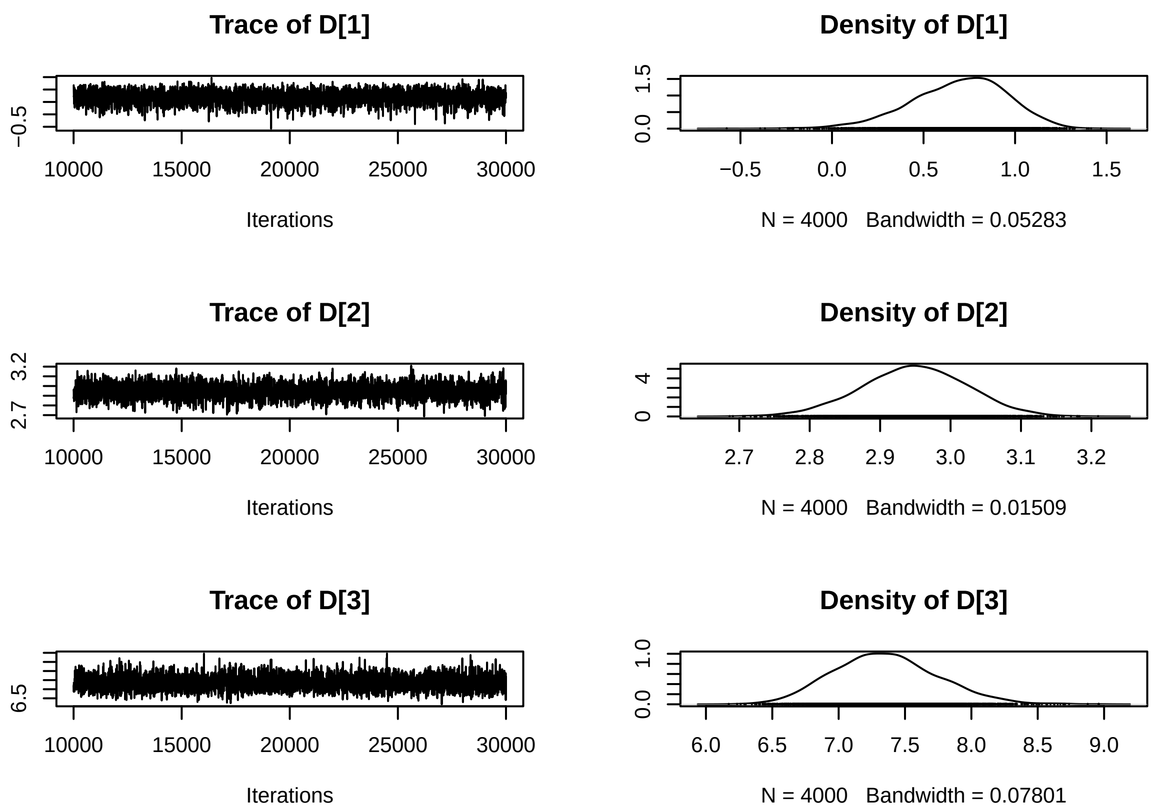
```r
# M1
model_string1 <- "model{

 # Likelihood
 for(i in 1:n){
   Y[i] ~ dnorm(mu[i], inv.var)
   mu[i] <- beta0 + beta1 * X[i]
 }

 #Priors
 beta0 ~ dnorm(0, 0.00001)
 beta1 ~ dnorm(0, 0.00001)
 inv.var ~ dgamma(0.01, 0.01)

 # Posterior preditive checks
 for(i in 1:n){
   Y1[i] ~ dnorm(mu[i], inv.var)
 }

 D[1] <- min(Y1[])
 D[2] <- mean(Y1[])
 D[3] <- max(Y1[])
}"

# M2
model_string2 <- "model{

 # Likelihood
 for(i in 1:n){
   Y[i] ~ dnorm(mu[i], inv.var)
   mu[i] <- beta0 + beta1 * X[i] + beta2 * X[i]^2
 }

 #Priors
 beta0 ~ dnorm(0, 0.00001)
 beta1 ~ dnorm(0, 0.00001)
 beta2 ~ dnorm(0, 0.00001)
 inv.var ~ dgamma(0.01, 0.01)

 # Posterior preditive checks
 for(i in 1:n){
   Y2[i] ~ dnorm(mu[i], inv.var)
 }

 D[1] <- min(Y2[])
 D[2] <- mean(Y2[])
 D[3] <- max(Y2[])
}"
```

3. Fit the two models

```
data <- list(Y = Y, X = X, n = n)

model1 <- jags.model(textConnection(model_string1),
                     data = data, n.chains = 1, quiet = TRUE)
update(model1, 10000, progress.bar = "none")
samps1 <- coda.samples(model1, variable.names = c("D"),
                       n.iter = 20000, thin = 5, progress.bar = "none")
plot(samps1)
```




```
D.m1 <- samps1[[1]]

model2 <- jags.model(textConnection(model_string2),
                     data = data, n.chains = 1, quiet = TRUE)
update(model2, 10000, progress.bar = "none")
samps2 <- coda.samples(model2, variable.names = c("D"),
                       n.iter = 20000, thin = 5, progress.bar = "none")
plot(samps2)
```

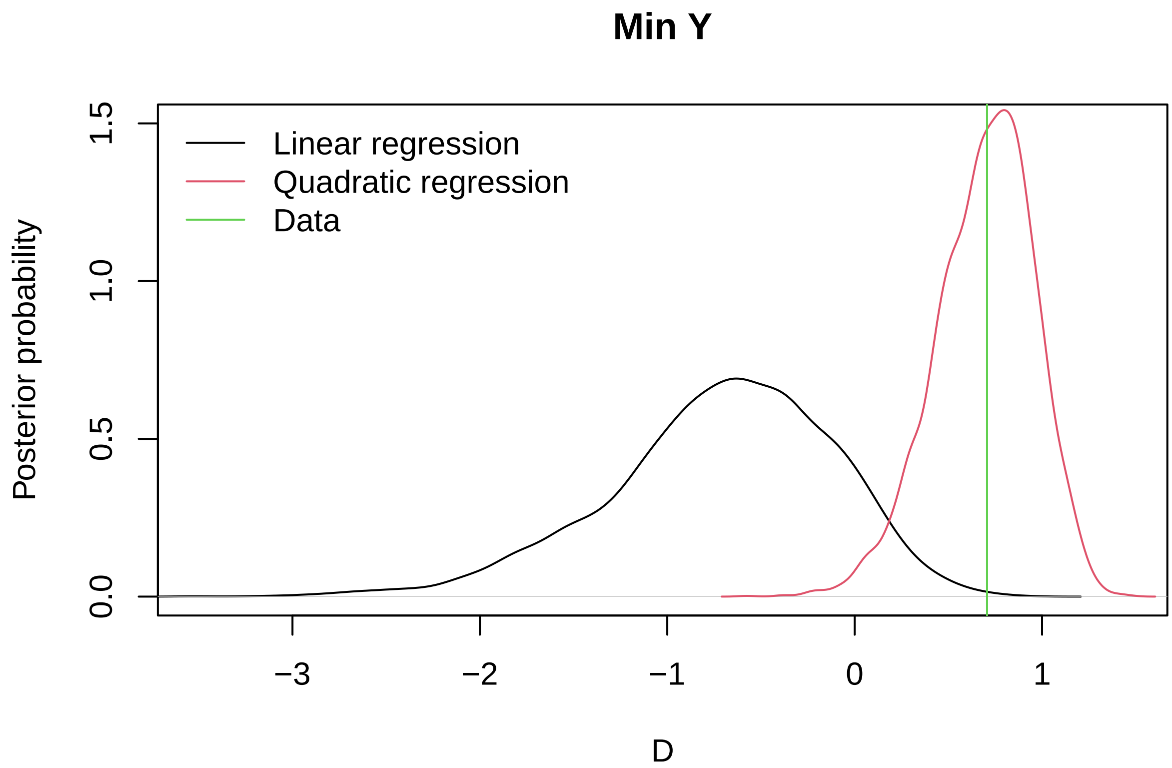


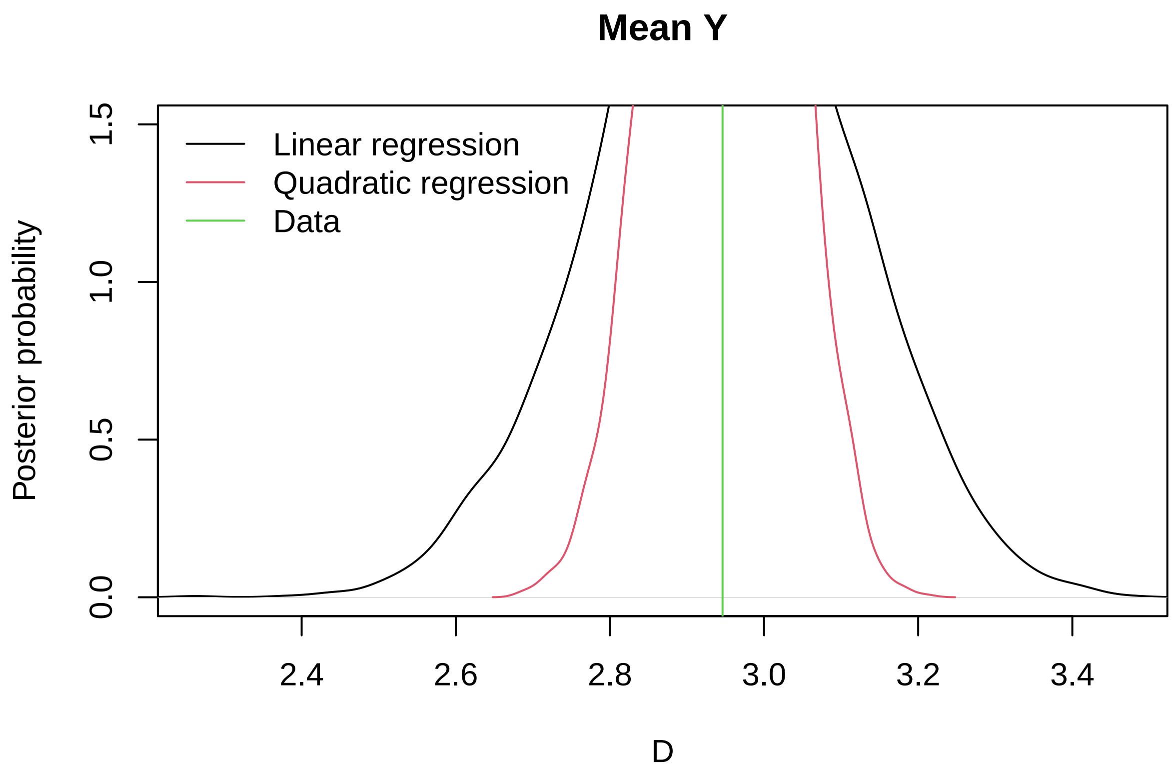

```
D.m2 <- samps2[[1]]
```

4. Compute the Bayesian p-values

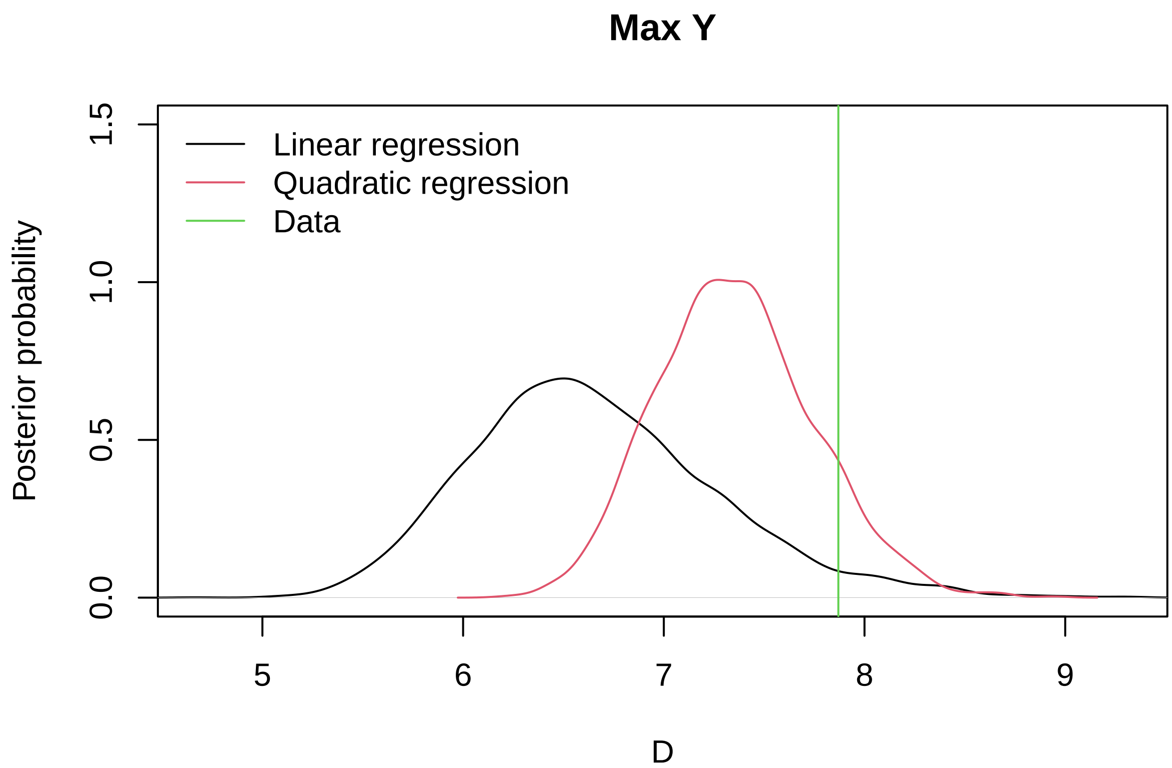
```
D0 <- c(min(Y), mean(Y), max(Y))
Dnames <- c("Min Y", "Mean Y", "Max Y")

# Compute the test stats for the models

pval1 <- pval2 <- rep(NA, 3)
names(pval1) <- names(pval2) <- c("Min Y", "Mean Y", "Max Y")

for(j in 1:3){
  plot(density(D.m1[ , j]), xlim = range(c(D0[j], D.m1[ , j], D.m2[ , j])),
       xlab = "D", ylab = "Posterior probability",
       main = Dnames[j], ylim = c(0, 1.5))
  lines(density(D.m2[ , j]), col = 2)
  abline(v = D0[j], col = 3)
  legend("topleft", c("Linear regression", "Quadratic regression", "Data"),
         lty = 1, col = 1:3, bty = "n")

  pval1[j] <- mean(D.m1[ , j] > D0[j])
```

```
  pval2[j] <- mean(D.m2[ , j] > D0[j])
}
```





Mean Y
Linear regression
Quadratic regression
Data
Posterior probability
0.0
0.5
1.0
1.5
2.4
2.6
2.8
3.0
3.2
3.4
D



5. Results

```
pval1
```

```
##   Min Y  Mean Y   Max Y
## 0.00125 0.50500 0.03975
```

```
pval2
```

```
##   Min Y  Mean Y   Max Y
## 0.51375 0.51450 0.10300
```