CrimeAnalysis

NarenSuri

November 28, 2016

```
# 9.3 Crime data Question
# a. Fit a regression model
library(MASS)
# in page # 159, its clearly given the algorithm to use the MontoCarlo simulation
# using the code discussed by the professor in the class
UsCrimeData<-UScrime
# assining the values from the data to the variables or selectign the relevant columns from the US crime data
UsCrimeResultantY<-UsCrimeData$y
CrimeDataToAnalyze<-UsCrimeData[,-16]
CrimeDataToAnalyze<-cbind(rep(1,47),CrimeDataToAnalyze)
CrimeDataToAnalyze<-as.matrix(CrimeDataToAnalyze)

# loaded all the required data above
# now lets initialize and set all the parameters required to start the iterative MC process
# initialize as shown in the text book
#############################
nu0<-2
s20=1
SigmaSquare=1
SamplingIters<-10000
Beta<-S2<-NULL
##############################
xtxinv<-solve(t(CrimeDataToAnalyze)%*%CrimeDataToAnalyze)
linearModelingResult<-lm(UsCrimeResultantY~CrimeDataToAnalyze[,-1])
# using code shared in text book and by Professor
SSRg<-t(UsCrimeResultantY)%*%UsCrimeResultantY-length(UsCrimeData$y)/(length(UsCrimeData$y)+1)*t(UsCrimeResultantY)%*%predic
t(linearModelingResult)
for(i in 1:SamplingIters){
  s2<-1/rgamma(n=1,nu0+length(UsCrimeData$y)/2,(nu0*s20-SigmaSquare+SSRg)/2)
  ValueOfbeta<-mvrnorm(n=1,mu=length(UsCrimeData$y)/(length(UsCrimeData$y)+1)*linearModelingResult$coefficients,length(UsCri
meData$y)/(length(UsCrimeData$y)+1)*s2*xtxinv)
  Beta<-rbind(Beta,ValueOfbeta)
  S2<-rbind(S2,s2)
}

########################
## Now working on the posterior samples
posteriorsample<-data.frame(Beta,S2)
colnames(posteriorsample)<-c("Beta0","Beta1","Beta2","Beta3","Beta4","Beta5","Beta6","Beta7","Beta8","Beta9","Beta10","Beta1
1","Beta12","Beta13","Beta14","Beta15","Sigma2")
```

```
posteriorMean<-colMeans(posteriorsample)
quantiles<-matrix(NA,15,2)
quantiles<-apply(posteriorsample,2,function(CrimeDataToAnalyze)quantile(CrimeDataToAnalyze,c(0.025,0.975)))
print(quantiles)
```

```
##           Beta0      Beta1     Beta2     Beta3     Beta4     Beta5
## 2.5%  -9084.060  0.2054869 -289.0855  5.722993 -2.093905 -34.50221
## 97.5% -2532.647 16.9374886  293.5188 30.745987 40.237493  12.59582
##           Beta6     Beta7     Beta8      Beta9     Beta10     Beta11
## 2.5%  -3.578822 -2.433715 -3.330872 -0.9194859 -14.067121 -0.2237992
## 97.5%  2.284444  5.684448  1.862107  1.7004010   2.705511 32.7217595
##          Beta12    Beta13     Beta14    Beta15   Sigma2
## 2.5%  -1.186508  2.323024 -9386.1591 -17.65502 31238.28
## 97.5%  2.988666 11.414119  -299.5392  10.42213 68425.54
```

```
print(posteriorMean)
```

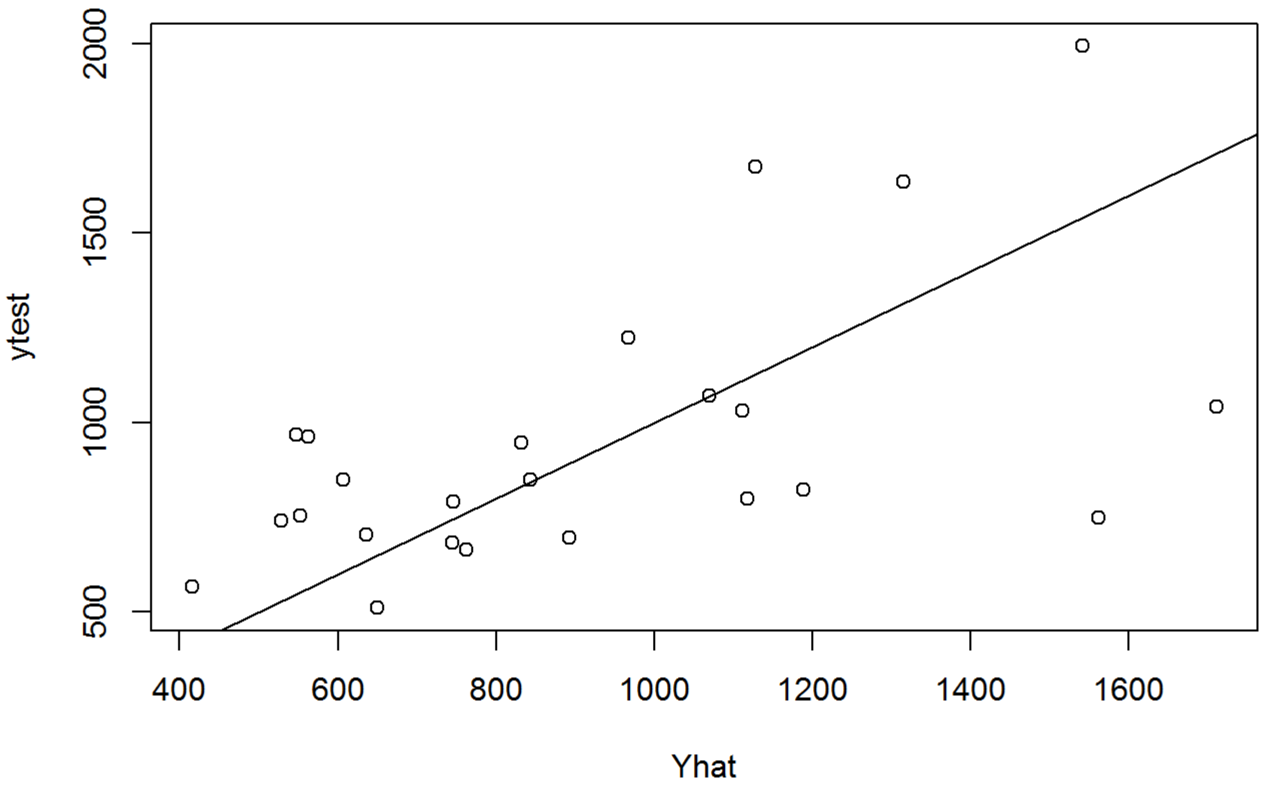
```
##         Beta0         Beta1         Beta2         Beta3         Beta4
## -5842.3570099     8.5544937    -0.4310164    18.4263013    19.0548710
##         Beta5         Beta6         Beta7         Beta8         Beta9
##   -10.9191383    -0.6434110     1.7072705    -0.6974008     0.4066948
##        Beta10        Beta11        Beta12        Beta13        Beta14
##    -5.7003581    16.3760101     0.9307654     6.8982190 -4795.8793622
##        Beta15        Sigma2
##    -3.5105993 46328.6092556
```

```
coffecients<-linearModelingResult$coefficients
print(summary(linearModelingResult))
```

```
##
## Call:
## lm(formula = UsCrimeResultantY ~ CrimeDataToAnalyze[, -1])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -395.74  -98.09   -6.69  112.99  512.67
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  -5984.2876  1628.3184  -3.675 0.000893 ***
## CrimeDataToAnalyze[, -1]M        8.7830     4.1714   2.106 0.043443 *
## CrimeDataToAnalyze[, -1]So      -3.8035   148.7551  -0.026 0.979765
## CrimeDataToAnalyze[, -1]Ed      18.8324     6.2088   3.033 0.004861 **
## CrimeDataToAnalyze[, -1]Po1     19.2804    10.6110   1.817 0.078892 .
## CrimeDataToAnalyze[, -1]Po2    -10.9422    11.7478  -0.931 0.358830
## CrimeDataToAnalyze[, -1]LF      -0.6638     1.4697  -0.452 0.654654
## CrimeDataToAnalyze[, -1]M.F      1.7407     2.0354   0.855 0.398995
## CrimeDataToAnalyze[, -1]Pop     -0.7330     1.2896  -0.568 0.573845
## CrimeDataToAnalyze[, -1]NW       0.4204     0.6481   0.649 0.521279
## CrimeDataToAnalyze[, -1]U1      -5.8271     4.2103  -1.384 0.176238
## CrimeDataToAnalyze[, -1]U2      16.7800     8.2336   2.038 0.050161 .
## CrimeDataToAnalyze[, -1]GDP      0.9617     1.0367   0.928 0.360754
## CrimeDataToAnalyze[, -1]Ineq     7.0672     2.2717   3.111 0.003983 **
## CrimeDataToAnalyze[, -1]Prob -4855.2658  2272.3746  -2.137 0.040627 *
## CrimeDataToAnalyze[, -1]Time    -3.4790     7.1653  -0.486 0.630708
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 209.1 on 31 degrees of freedom
## Multiple R-squared:  0.8031, Adjusted R-squared:  0.7078
## F-statistic: 8.429 on 15 and 31 DF,  p-value: 3.539e-07
```

```
##############################################################################################################

#Question part 2 - b - 1
SizeOfSamlple<-0.5*length(UsCrimeData$y)
## set the seed to make your partition reproductible
set.seed(145)
trainDataindexValues <- sample(seq_len(length(UsCrimeData$y)), size = SizeOfSamlple)
train <- UsCrimeData[trainDataindexValues, ]
test <- UsCrimeData[-trainDataindexValues, ]
trainValueOfY<-as.matrix(train$y)
Xtrain<-as.matrix(train[,-16])
ytest<-as.matrix(test$y)
Xtest<-as.matrix(test[,-16])
Xtest<-cbind(rep(1,nrow(Xtest)),Xtest)
linear_model<-lm(trainValueOfY~Xtrain)
B<-linear_model$coefficients
Yhat<-Xtest%*%B
plot(Yhat,ytest)
abline(0,1)
```



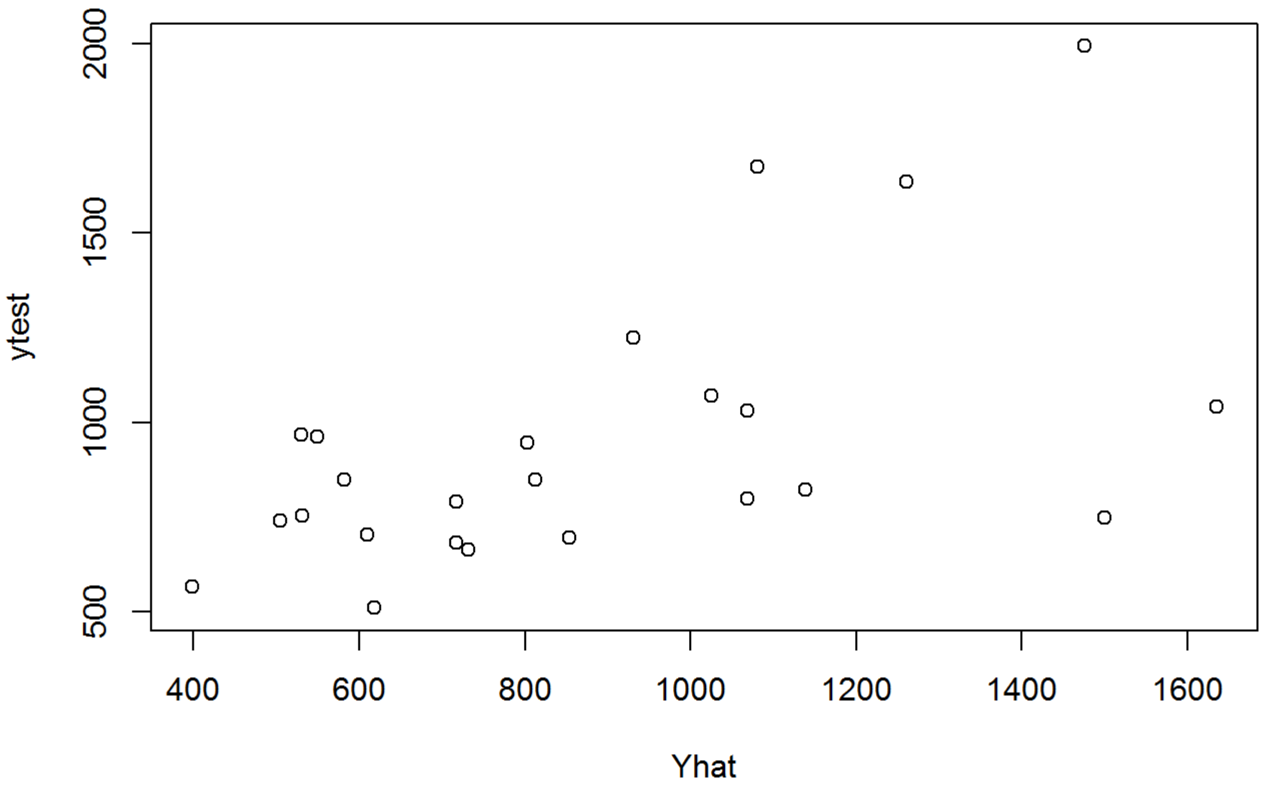

```
perror<-(1/length(ytest))*sum((ytest-Yhat)^2)
print(perror)
```

```
## [1] 108998
```

```
##########################################################################################

##Question part 2 - b - 2
train <- UsCrimeData[trainDataindexValues, ]
test <- UsCrimeData[-trainDataindexValues, ]
trainValueOfY<-as.matrix(train$y)
UsCrimeResultantY<-trainValueOfY
Xtrain<-as.matrix(train[,-16])
Xtrain<-cbind(rep(1,nrow(Xtrain)),Xtrain)
CrimeDataToAnalyze<-Xtrain
ytest<-as.matrix(test$y)
Xtest<-as.matrix(test[,-16])
Xtest<-cbind(rep(1,nrow(Xtest)),Xtest)

nu0<-2
s20 <- SigmaSquare<-1
SamplingIters<-10000
Beta<-S2<-NULL
n<-length(trainValueOfY)
xtxinv<-solve(t(CrimeDataToAnalyze)%*%CrimeDataToAnalyze)
linearModelingResult<-lm(UsCrimeResultantY~CrimeDataToAnalyze[,-1])
SSRg<-t(UsCrimeResultantY)%*%UsCrimeResultantY-length(trainValueOfY)/(length(trainValueOfY)+1)*t(UsCrimeResultantY)%*%predic
t(linearModelingResult)
for(i in 1:SamplingIters){
  s2<-1/rgamma(n=1,nu0+length(trainValueOfY)/2,(nu0*s20-SigmaSquare+SSRg)/2)
  ValueOfbeta<-mvrnorm(n=1,mu=length(trainValueOfY)/(length(trainValueOfY)+1)*linearModelingResult$coefficients,length(train
ValueOfY)/(length(trainValueOfY)+1)*s2*xtxinv)
  Beta<-rbind(Beta,ValueOfbeta)
  S2<-rbind(S2,s2)
}
posteriorsample<-data.frame(Beta)
colnames(posteriorsample)<-c("Beta0","Beta1","Beta2","Beta3","Beta4","Beta5","Beta6","Beta7","Beta8","Beta9","Beta10","Beta1
1","Beta12","Beta13","Beta14","Beta15")
posteriorMean<-colMeans(posteriorsample)
posteriorMean<-as.matrix(posteriorMean)
Yhat<-Xtest%*%posteriorMean
plot(Yhat,ytest)
```




```
perror<-(1/length(ytest))*sum((ytest-Yhat)^2)
print(perror)
```

```
## [1] 107600.3
```