Data Mining Homework 5

Ashish Patidar

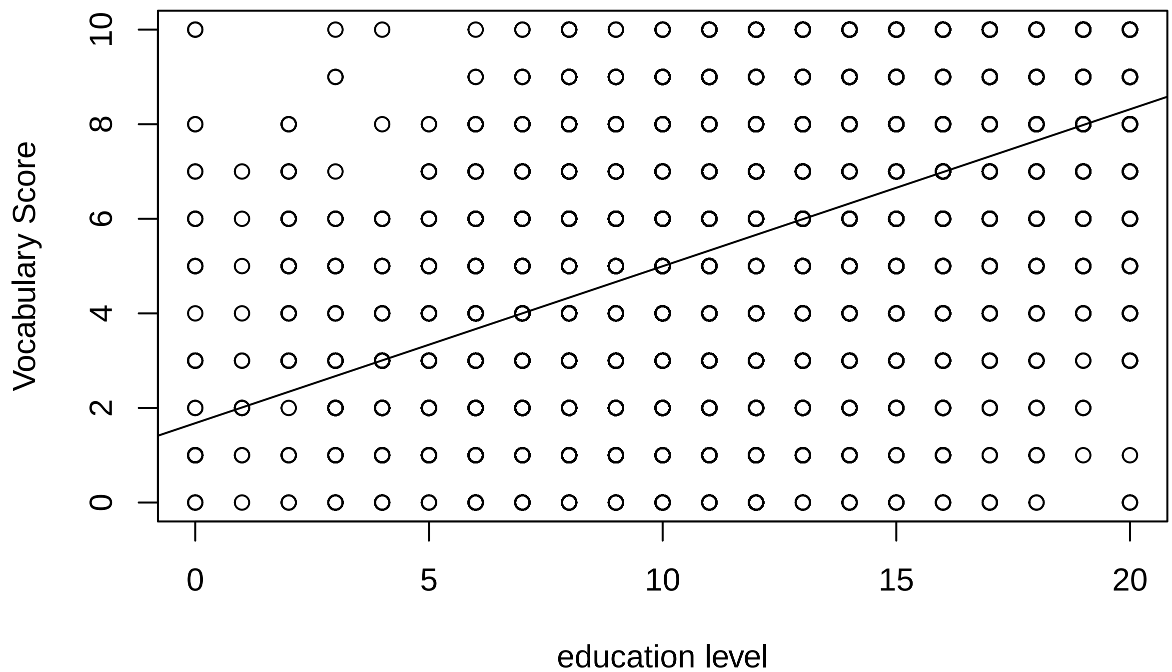
Answer 1

```
#Reading data from file
data_1 = read.csv("/Users/ashish/Downloads/Vocab.csv")

#defining x and y
x = data_1$education
y = data_1$vocabulary
n = length(x)
X = cbind(x,rep(1,n))
a = solve(t(X) %*% X , t(X) %*% y)
cat("a and b are ", a[1], a[2], "\n")
```

```
## a and b are  0.3318736 1.677939
```

```
#plot of line to see the trend
plot(x,y,xlab = "education level",ylab="Vocabulary Score")
abline(a[2],a[1])
```



If we look at the above line we can say with education level vocabulary score is improving, therefore, we can say that people with more education tend to have larger vocabularies. As our predictor variable x is education level and response y is vocabulary score, so using our estimates of a and b we can say that: vocabulary score = 0.3319*education level + 1.6779

Answer 2

```r
data_2 = read.csv("/Users/ashish/Downloads/ais.csv")

#defining all the variables to solve normal equations
X = as.matrix(data_2[,3:12])
y = data_2[,2]
a = solve(t(X) %*% X , t(X) %*% y)

#predicting values from the model we created
y_hat = X %*% a

#sum of squared error calculation
error = (y - y_hat)^2
SSE = sum(error)

#loop to check the most important variable
SSE_list = rep(0,10)

for (i in 1:10){
  X1 = X[,-(i)]
  y = data_2[,2]
  a = solve(t(X1) %*% X1 , t(X1) %*% y)
  y_hat = X1 %*% a
  error = (y - y_hat)^2
  SSE = sum(error)
  SSE_list[i] = SSE
}
cat("SSE is ",SSE, "\n")
```

```
## SSE is  5.914183
```

```r
imp_col = names(data_2[which.max(SSE_list) + 2])
cat("Most important column is :", imp_col,"\n")
```

```
## Most important column is : hc
```

From above output we can see that SSE is 5.9142 and hc is the most important column in predicting rce.

Answer 3

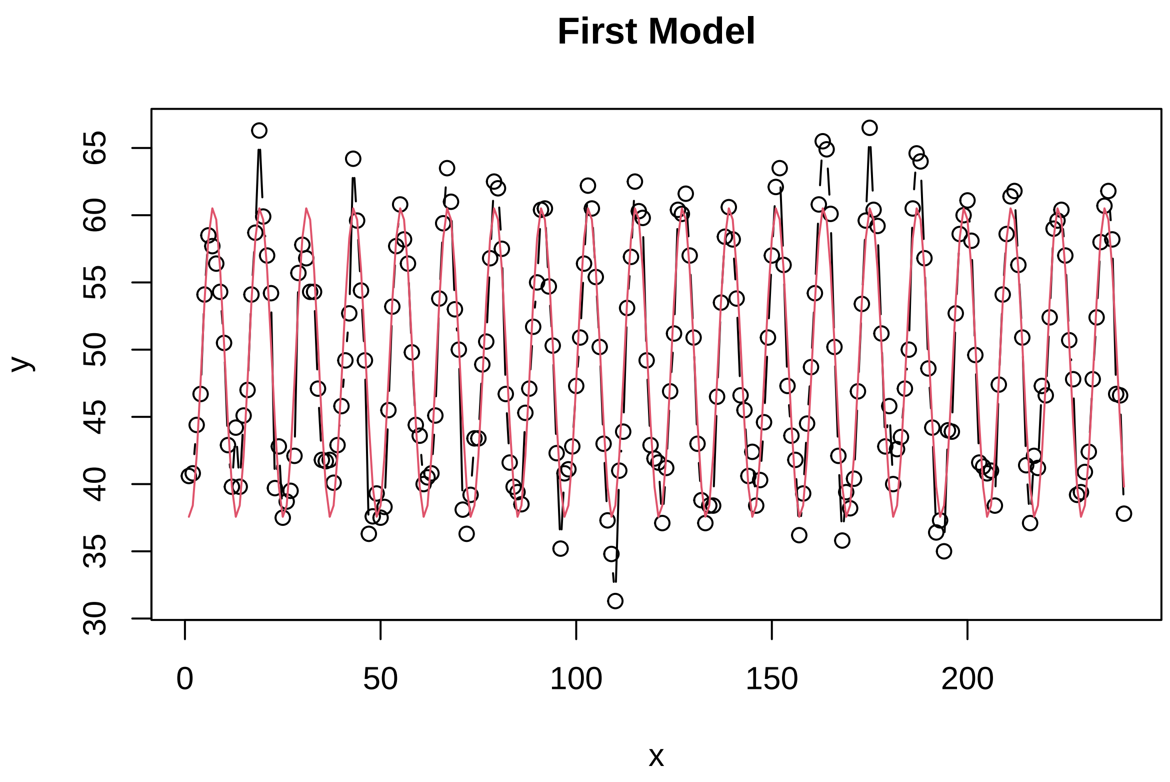
```
#using nottingham beer sales data and defining variables
data(nottem)
y = nottem
n = length(y)
x = 1:n

plot(x,y, type = "b", main = "First Model")

#Question 3b)
x1 = cos((2*pi*x)/12)
x2 = sin((2*pi*x)/12)
X1 = cbind(x1,x2,rep(1,n))
a1 = solve(t(X1) %*% X1 , t(X1) %*% y)
y_hat1 = X1%*%a1
cat("a, b and c are ", a1[1], a1[2],a1[3], "\n")
```

```
## a, b and c are  -9.240921 -6.940906 49.03958
```

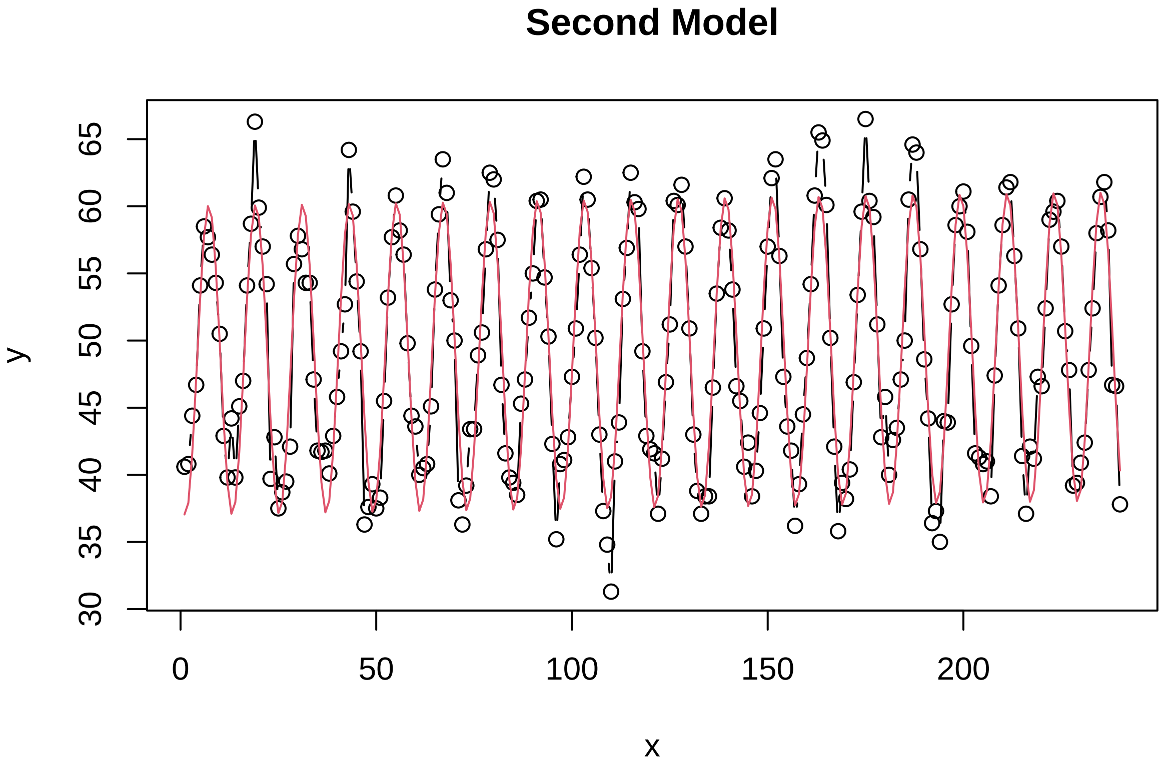
```
#Model fitting
lines(x, y_hat1, col = 2)
```




```
#Question 3c)
x1 = cos((2*pi*x)/12)
x2 = sin((2*pi*x)/12)
X2 = cbind(x1,x2,rep(1,n),x)
a2 = solve(t(X2) %*% X2 , t(X2) %*% y)
y_hat2 = X2%*%a2
cat("a, b, c and d are ", a2[1], a2[2],a2[3],a2[4], "\n")
```

```
## a, b, c and d are  -9.245314 -6.924513 48.51032 0.004392239
```

```r
#Model fitting
plot(x,y, type = "b", main = "Second Model")
lines(x, y_hat2, col = 2)
```



If we look at the coefficients we got, for d coefficient is 0.0044 which is directly related to number of days, so by looking at that we can say that there is very little increase in sales of beers.

Answer 4

```
#reading all the files
data_4_1 = read.table("/Users/ashish/Downloads/pred1.dat.txt")
data_4_2 = read.table("/Users/ashish/Downloads/pred2.dat.txt")
y_1 = read.table("/Users/ashish/Downloads/resp1.dat.txt")
y_2 = read.table("/Users/ashish/Downloads/resp2.dat.txt")

#defining variables
X1_1st = as.matrix(data_4_1[1:(nrow(data_4_1)/2),])
X2_1st = as.matrix(data_4_2[1:(nrow(data_4_2)/2),])

X1_2nd = as.matrix(data_4_1[(nrow(data_4_1)/2 + 1):nrow(data_4_1),])
X2_2nd = as.matrix(data_4_2[(nrow(data_4_2)/2 + 1):nrow(data_4_2),])

y1_1st = y_1[1:(nrow(y_1)/2),]
y1_2nd = y_1[(nrow(y_1)/2 + 1):nrow(y_1),]

y2_1st = y_2[1:(nrow(y_2)/2),]
y2_2nd = y_2[(nrow(y_2)/2 + 1):nrow(y_2),]

#Question 4a)
#solving normal equations for estimation of parameters
a1 = solve(t(X1_1st) %*% X1_1st , t(X1_1st) %*% y1_1st)
a2 = solve(t(X2_1st) %*% X2_1st , t(X2_1st) %*% y2_1st)

y1_hat = X1_2nd %*% a1
y2_hat = X2_2nd %*% a2

#Question 4b)
#calculation of SSE
SSE_1 = sum((y1_2nd - y1_hat)^2)
SSE_2 = sum((y2_2nd - y2_hat)^2)

cat("For 1st dataset SSE is", SSE_1, "\n", "For second dataset SSE is", SSE_2, "\n")
```

```
## For 1st dataset SSE is 5.721507
##  For second dataset SSE is 32984664
```

Answer 5

```
#Question 5a)

p = ncol(data_4_1)
used = rep(FALSE,p)      # initially all varaibles available for selection
var = rep(0,p)           # var[j] will be variable chosen in jth round
bestsse = rep(10000000,p)      # bestsse[j] will be best sse from jth round

min_col = rep(0,3)

#loop to apply forward selection
for (j in 1:p)  {        # choose 1 variable each time through this loop
  for (i in which(used == FALSE)) {
    used[i] = TRUE
    XX = X1_1st[,used]       # take the "used" columns = used variables
    a = solve(t(XX) %*% XX , t(XX) %*% y1_1st)
    yhat = XX %*% a
    error = y1_2nd-yhat
    sse = sum(error*error)
    if (sse < bestsse[j]) { # if we find a better sse, take it
      bestsse[j] = sse
      var[j] = i
    }
    used[i] = FALSE

  }
  used[var[j]] = TRUE    # claim the best variable for future iterations of loop
}

#loop to extract top 3 variables (backward selection)
for (i in 1:3){
  min_col[i] = var[length(var)]
  var = var[-length(var)]
}
cat("Three best predictors are:", names(data_4_1[min_col]), "\n")
```

```
## Three best predictors are: V21 V47 V16
```

```
#Question 5b)

X = cbind(data_4_1[1:(nrow(data_4_1)/2),min_col[1]],data_4_1[1:(nrow(data_4_1)/2),min_col[2]],data_4_1[
a = solve(t(X) %*% X , t(X) %*% y1_1st)

X1 = cbind(data_4_1[(nrow(data_4_1)/2 + 1):nrow(data_4_1),min_col[1]], data_4_1[(nrow(data_4_1)/2 + 1):
y_hat = X1 %*% a
SSE = sum((y1_2nd - y_hat)^2)
cat("SSE with all the columns was: ", SSE_1, "\nSSE with our model is: ", SSE)
```

```
## SSE with all the columns was:  5.721507
## SSE with our model is:  5.099969
```

It is clear from the above observations that we are getting better SSE for our model, it is because these three are the important predictors and rest of them are just adding small amount of noise to the model because of which both the SSE's are close to each other.

Answer 6

```r
#Question 6a)
p = ncol(X2_1st)
lambda = 20
a = solve(t(X2_1st) %*% X2_1st + lambda*diag(p), t(X2_1st) %*% y2_1st)
y_hat_6 = X2_2nd %*% a

#Question 6b)

SSE_6 = sum((y2_2nd - y_hat_6)^2)
cat("SSE with plain regression was: ", SSE_2, "\n SSE with ridge regression is: ", SSE_6, "\n")
```

```
## SSE with plain regression was:  32984664
##  SSE with ridge regression is:  35918.6
```

```r
#Question 6c)

lam_choices = seq(-100,100, by = 2)
error = rep(0,length(lam_choices))

for (i in 1:length(lam_choices)) {
  lambda = lam_choices[i]
  a = solve(t(X2_1st) %*% X2_1st + lambda*diag(p), t(X2_1st) %*% y2_1st);   # ridge regression solves d
  #  print(a)
  yhat = X2_2nd %*% a;
  error[i] = sum((y2_2nd-yhat)^2)
}
cat("Best Lambda choice is: ", lam_choices[which.min(error)], "with SSE = ", min(error), "\n")
```

```
## Best Lambda choice is:  6 with SSE =  30878.57
```

Answer 7

```r
#reading data from file

data_7 = read.csv("/Users/ashish/Downloads/time_series.dat", header = F)

#defining variables
x1 = as.matrix(data_7[2:(nrow(data_7) - 1),])
x2 = as.matrix(data_7[1:(nrow(data_7) - 2),])
y = as.matrix(data_7[3:(nrow(data_7)),])
n = nrow(data_7)
X = cbind(x1, x2)

#estimating alpha1 and alpha2
a = solve(t(X) %*% X , t(X) %*% y)

y_hat = X %*% a
SSE = (y - y_hat)^2

#calculating variance of SSE
cat("alpha1 = ", a[1], "\nalpha2 = ", a[2], "\nvariance of error = ", var(SSE), "\n")
```

```
## alpha1 =  0.990185
## alpha2 =  -0.9383054
## variance of error =  1.182408e-05
```