MATH 3333 3.0 - Winter 2021-2022

Assignment 1

(Due Date: Feb 4, 2022)

For all the questions involving R programming, please submit your R code and attach the screenshot of the R output.

Question 1: Please download the bike sharing data set from the following website: https://archive.ics.uci.edu/ml/datasets/bike+sharing+dataset. (I also include the data set on crowdmark.) This dataset contains the hourly and daily count of rental bikes between years 2011 and 2012 in Capital bikeshare system with the corresponding weather and seasonal information. There are two data sets and we will use the day.csv. Import the data into R and perform the following exploratory analysis.
a) The variable "registered" records the number of registered users used the bike sharing service on a particular day. Please provide the mean value of the variable "registered" for each day of the week.

b) Plot the conditional density plot of the variable "registered" conditional on each month of the year.

c) Produce a two-dimensional levelplot of the variable "registered" against the combination of temperature (variable "temp") and humidity (variable "hum").

Question 2: Perform linear regression model on the bike sharing data set from Question 2.

a. Provide the summary result of the regression model with "registered" as the response variable and "temp", "hum" as the predictors.

b. What other predictors do you think might be important for the modelling of the variable "registered"? Please construct another linear model including more predictors and provide the summary result of the second model.

c. Perform 100 times of 5-fold cross validation and each time you randomly partition the dataset into five equal parts. You will use 80% of the data as the training data and 20% as the validating data. For models in a) and b), calculate the total sum of squared prediction error divived by the size of the validation data and by the number of cross-validations. Which model has better predictive power?

Question 3: In the following marketing set, we have 9 years with the sales in 10 million euro and the advertising expenditure in million euro.

Year	Sales	Advertisement
1	34	23
2	56	36
3	65	45
4	86	52
5	109	53
6	109	58
7	122	63
8	124	68
9	131	70

a) Formulate the response vector Y, which has nine entries.

b) Formulate the data matrix of X, the first column should be all ones corresponding to the intercept, and the second column should be the predictors. The dimension of X should be 9×2.

c) Write R code to compute X^tX.

d) Write R code to compute $\theta = (X^tX)^-X^tY$. This is the estimated linear regression coefficient of the linear model with Y as the response and X as the data matrix.

e)Run the linear regression using Y as the response and X as the predictor using lm command in R and compare the output with your own calculation.

f) Now two additional data points arrived. They are Year 10, Sales 96, and Advertisement 53; Year 11, Sales 107, and Advertisement 63. Please use the online algorithm to update the linear model. Use the two new observations together to perform the sequential learning and update the model using stochastic gradient descent algorithm using the learning rate $\lambda = 0.01$. Note here in the updating scheme $\hat{\theta}^{new} = \hat{\theta}^{old} - \lambda \nabla E_n$, the term E_n will be the sum of the squared prediction errors over the two new observations:

$$E_n = (y_{10} - \hat{y}_{10})^2 + (y_{11} - \hat{y}_{11})^2.$$

In this example, we implement the online algorithm when the new data come in batches, not one at a time.

Question 4: Consider a data set with the response vector $Y = (y_1, \ldots, y_n)^t$ and the data matrix

$$\begin{pmatrix} 1 & x_{11} & x_{12} \\ . & . & . \\ . & . & . \\ 1 & x_{n1} & x_{n2} \end{pmatrix}.$$

We model the relationship between X and Y using the linear regression model: $y_i = \theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + \epsilon_i$, $i = 1, \ldots, n$, where $\epsilon \sim N(0, \sigma^2)$. Let the parameter vector be denoted as $\theta = (\theta_0, \theta_1, \theta_2)^t$. We wish to minimize the sum of squared residuals: $SSE =$

$\sum_{i=1}^{n}(y_i - (\theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2}))^2$. Let the fitted value be denoted as $\hat{y}_i = \theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2}$, and let the fitted value vector be denoted as $\hat{Y}$.

a) Show that $SSE = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$.

b) Show that $SSE = (Y - \hat{Y})^t(Y - \hat{Y})$.

c) Show that $\hat{Y} = X\theta$.

d) Simplify the derivative equation $\frac{\partial SSE}{\partial \theta} = 0$.

e) Find the solution of θ which solves the equation in part d.

Question 5: Analyze the QSAR fish toxicity data set from the site: https://archive.ics.uci.edu/ml/datasets/QSAR+fish+toxicity. Perform variable selection on the six predictors using the lasso package.

a) Based on the output of "lars", please provide the sequence of candidate models. For example, the first model is $\{X_5\}$, the second model is $\{X_5, X_3\}$ and the third model is $\{X_5, X_3, X_{10}\}$, etc.

b) Use the cross validation method, select the best value for the fraction s based on the plot of cross validation error againt the fraction s. The fraction s measures the ratio of the L_1 norm of the penalized estimate over the L_1 norm of the regular penalized estimate.

c) Use the optimum s you select, perform the penalized regression and output the optimum model and the estimated coefficients.

Question 6: Simulate a data set with 100 observations $y_i = 30 + 5x + 2x^2 + 3x^3 + \epsilon_i$, where ϵ_i follows independent normal distribution $N(0, 1)$.
a) Perform polynomial regression on your simulated data set and using $x, I(x^2), I(x^3)$ as the predictors. Compare the estimated coefficients with the true model and report the R-square.
b) Formulate the design matrix of this regression and write down the first two rows of the design matrix based on your data set.
c) Perform polynomial regression on your simulated data set and using $x, I(x^2), I(x^3), I(x^4)$ as the predictors. Compare the estimated coefficients with the true model and report the R-square. Is the R-square increased or decreased compared to the model in part (a)? Explain why?

3333 Assignment 1

Ravish Kamath: 213893664

02/04/2022

```
day = read.csv('/Users/ravishkamath/Desktop/University/2. York Math/1 MATH/Statistics /MATH 3333/3. Ass
fishToxicity = read.csv('/Users/ravishkamath/Desktop/University/2. York Math/1 MATH/Statistics /MATH 33
library(lars)
```

```
## Loaded lars 1.2
```

```
library(lattice)
library(lars)
library(locfit)
```

```
## locfit 1.5-9.4    2020-03-24
```

```
library(knitr)
library(ggplot2)
```

Question 1

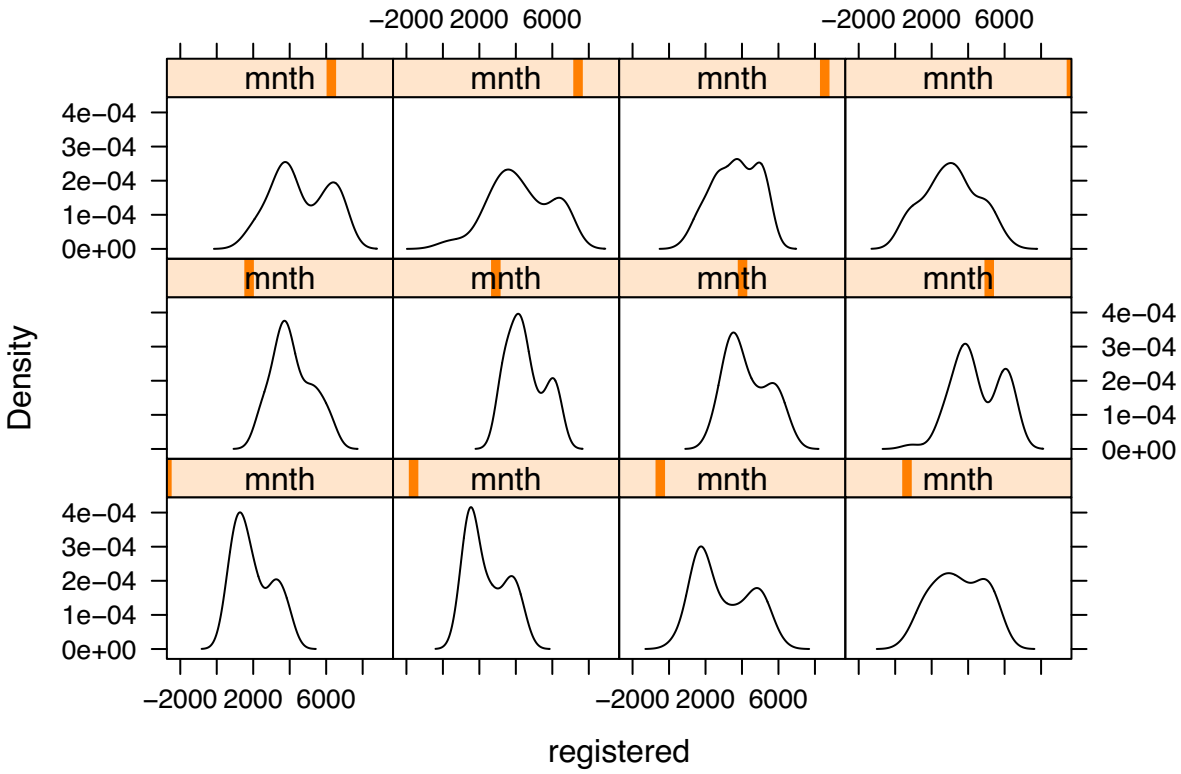
Part A

```
registeredMeans = aggregate(day$registered, by = list(day$weekday),
                            FUN = function(x) {round(mean(x), digits = 0)})
days = matrix(c('Sunday', 'Monday', 'Tuesday', 'Wednesday',
                'Thursday', 'Friday', 'Saturday'), byrow = T)
means = cbind(days, registeredMeans)
names(means) = c('Days', 'Days ID', 'Registered Mean')
means
```

```
##        Days Days ID Registered Mean
## 1    Sunday       0            2891
## 2    Monday       1            3664
## 3   Tuesday       2            3954
## 4 Wednesday       3            3997
## 5  Thursday       4            4076
## 6    Friday       5            3938
## 7  Saturday       6            3085
```

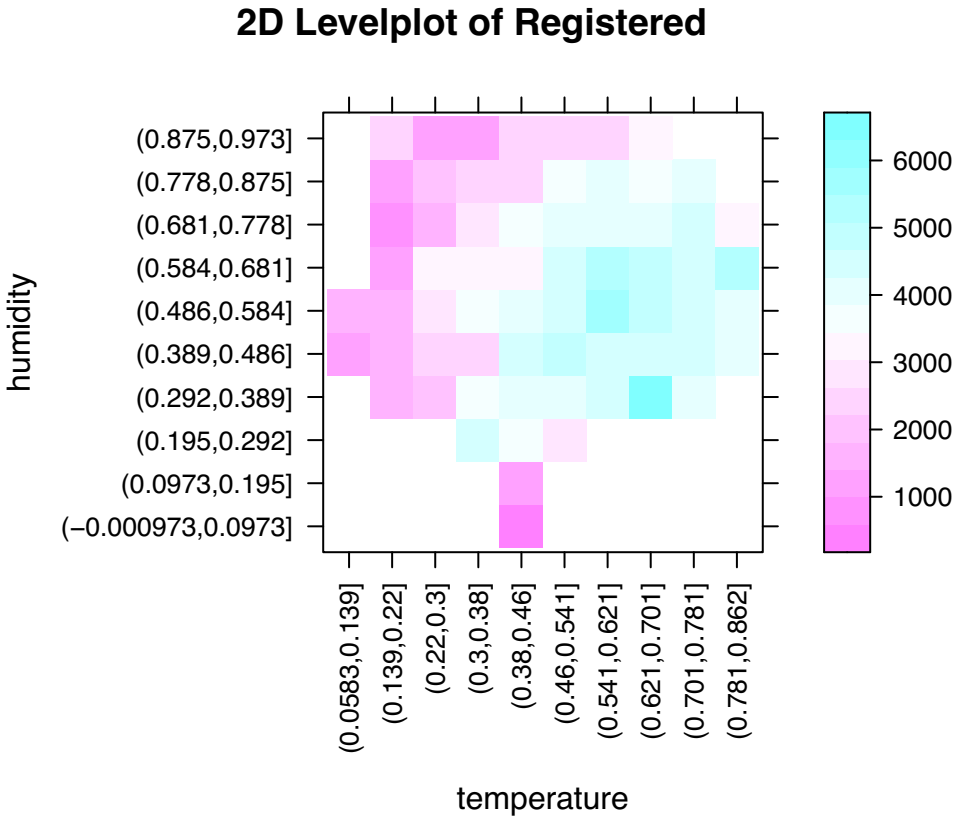
Part B

```
library(lattice)
densityplot(~registered|mnth, data = day, plot.points = FALSE, col = "black")
```


−2000 2000 6000
−2000 2000 6000
mnth
mnth
mnth
mnth
mnth
mnth
mnth
mnth
mnth
mnth
mnth
mnth
4e−04
3e−04
2e−04
1e−04
0e+00
Density
−2000 2000 6000
−2000 2000 6000
registered

Part C

```
df_brakes = tapply(day$registered,
                   INDEX = list(cut(day$temp, breaks = 10),
                                cut(day$hum, breaks = 10)), FUN = mean, na.rm = TRUE)
levelplot(df_brakes, scales = list(x = list(rot = 90)), main = '2D Levelplot of Registered',
          xlab = 'temperature', ylab = 'humidity')
```



Question 2

Part A

```
mod = lm(registered ~ temp + hum, data = day)
summary(mod)
```

```
##
## Call:
## lm(formula = registered ~ temp + hum, data = day)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3687.4  -994.5  -177.7   968.0  3170.3
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2405.1      239.4  10.046  < 2e-16 ***
## temp          4778.5      263.1  18.162  < 2e-16 ***
## hum          -1777.6      338.1  -5.257 1.93e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1291 on 728 degrees of freedom
## Multiple R-squared:  0.3175, Adjusted R-squared:  0.3156
## F-statistic: 169.3 on 2 and 728 DF,  p-value: < 2.2e-16
```

Part B

```
mod1 = lm(registered ~ temp + hum + workingday + weathersit, data = day)
summary(mod1)
```

```
##
## Call:
## lm(formula = registered ~ temp + hum + workingday + weathersit,
##     data = day)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2907.45  -960.55   -74.98   901.86  2799.40
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1948.26     229.30   8.497  < 2e-16 ***
## temp         4340.16     251.88  17.231  < 2e-16 ***
## hum          -584.22     397.70  -1.469    0.142
## workingday    971.65      95.51  10.173  < 2e-16 ***
## weathersit   -530.27     104.10  -5.094 4.47e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1196 on 726 degrees of freedom
## Multiple R-squared:  0.4161, Adjusted R-squared:  0.4129
## F-statistic: 129.3 on 4 and 726 DF,  p-value: < 2.2e-16
```

Part C

```
### CV for part a)
totalCVerror<-0
for ( j in 1:1){
  training<-sample(1:731,584)
  trainingset<-day[training,]
  testingset<-day[-training,]
  ###fill in the codes to get the predictions errors
  ### update the totalCVerror
  model<-lm(registered~temp+hum, data=trainingset)
  prediction<-predict(model, new = testingset)
  errors<-sum((testingset$registered - prediction)^2)
  totalCVerror<-totalCVerror+errors
}
averageCVerrors_modA<-totalCVerror/100/147
averageCVerrors_modA
```

```
## [1] 16434.51
```

```
### CV for part b)
totalCVerror<-0
for ( j in 1:1){
  training<-sample(1:731,584)
  trainingset<-day[training,]
  testingset<-day[-training,]
  ###fill in the codes to get the predictions errors
  ### update the totalCVerror
  model<-lm(registered~temp+hum + workingday + weathersit, data=trainingset)
  prediction<-predict(model, new = testingset)
  errors<-sum((testingset$registered - prediction)^2)
  totalCVerror<-totalCVerror+errors
}
averageCVerrors_modB<-totalCVerror/100/147
averageCVerrors_modB
```

```
## [1] 14343.96
```

```
#In this case, the model used in part B, has a better predictive power,
#since it has less errors.
```

Question 3

Part A

```
Y = c(34, 56, 65, 86, 109, 109, 122, 124, 131)
matrix(Y, ncol = 1, byrow = T)
```

```
##       [,1]
##  [1,]   34
##  [2,]   56
##  [3,]   65
##  [4,]   86
##  [5,]  109
##  [6,]  109
##  [7,]  122
##  [8,]  124
##  [9,]  131
```

Part B

```
advertisement = c(23, 36, 45, 52, 53, 58, 63, 68, 70)
ones = c(rep(1,9))
X = cbind(ones, advertisement)
X
```

```
##       ones advertisement
##  [1,]    1            23
##  [2,]    1            36
##  [3,]    1            45
##  [4,]    1            52
##  [5,]    1            53
##  [6,]    1            58
##  [7,]    1            63
##  [8,]    1            68
##  [9,]    1            70
```

Part C

```
XtransX = t(X)%*%X
XtransX
```

```
##                ones advertisement
## ones              9           468
## advertisement   468         26220
```

Part D

```
XtransY = t(X)%*%Y
theta = (solve(XtransX))%*%XtransY
theta
```

```
##                      [,1]
## ones          -20.550602
## advertisement   2.181529
```

Part E

```
mod = summary(lm(Y~X))
mod
```

```
##
## Call:
## lm(formula = Y ~ X)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.618  -3.793  -1.156   4.375  13.930
##
## Coefficients: (1 not defined because of singularities)
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -20.5506    10.2415  -2.007   0.0848 .
## Xones               NA         NA      NA       NA
## Xadvertisement   2.1815     0.1897  11.497 8.47e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.236 on 7 degrees of freedom
## Multiple R-squared:  0.9497, Adjusted R-squared:  0.9425
## F-statistic: 132.2 on 1 and 7 DF,  p-value: 8.47e-06
```

```
#As you can see, when using the lm function vs. manually solving the
#Least Square method, we end up having the same model and coefficients.
```

Part F

```
#Please check the handwritten notes
```

Question 3

f) Based on part e, we got our predicted model to be:

$\hat{y}_i = -20.5506 + 2.1815x_i$

Using the Stochastic Gradient Descent: $y_n = 96$ $x_n = 53$ $\hat{y}_{n+1} = 107$ $x_{n+1} = 63$

$$(y_n - \hat{y}_n) = (96 - (-20.5506 + 2.1815(53)))$$
$$= 0.9311$$
$$(y_{n+1} - \hat{y}_{n+1}) = (107 - (-20.5506 + 2.1815(63)))$$
$$= -9.8839$$

$$\nabla E_n + \nabla E_{n+1}$$

$$= -2(y_n - x_n^t\theta)\begin{pmatrix}1\\x_1\end{pmatrix} - 2(y_n - x_{n+1}^t\theta)\begin{pmatrix}1\\x_1\end{pmatrix}$$

$$= -2(0.9311)\begin{pmatrix}1\\53\end{pmatrix} - 2(-9.8839)\begin{pmatrix}1\\63\end{pmatrix}$$

$$= -1.8622\begin{pmatrix}1\\53\end{pmatrix} + 19.7678\begin{pmatrix}1\\63\end{pmatrix}$$

$$= \begin{pmatrix}17.9056\\1146.6748\end{pmatrix}$$

$$\theta_0^{(new)} = -20.5506 - 0.01(17.9056)$$
$$= -20.729656$$
$$\theta_1^{(new)} = 2.1815 - 0.01(1146.6748)$$
$$= -9.285248$$

Hence the new model is: $\hat{y}_i = -20.7297 - 9.2852x_i$

Question 4

a) Since it is given in the question that $SSE = \sum_{i=1}^{n} (y_i - (\theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2}))^2$ and $\hat{y}_i = \theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2}$, then

$$SSE = \sum_{i=1}^{n} (y_i - (\theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2}))^2$$

$$= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

b) $$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$= (y_i - \hat{y}_i)^T (y_i - \hat{y}_i) \quad \text{where } (y_i - \hat{y}_i) \text{ is a vector}$$

We can rewrite y_i in terms of vector $Y = (y_1, \ldots, y_n)^t$ and $\hat{Y} = \hat{y}_i$. Hence we can remove

$$= (Y - \hat{Y})^T (Y - \hat{Y})$$

c) To show $\hat{Y} = X\theta$

Let $X = \begin{pmatrix} 1 & x_{11} & x_{12} \\ \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} \end{pmatrix}$ and $\theta = (\theta_0, \theta_1, \theta_2)^t$

$n \times 3$

$$= \begin{pmatrix} 1 & x_{11} & x_{12} \\ \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} \end{pmatrix} \begin{pmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{pmatrix}$$

$$= \begin{pmatrix} \theta_0 + \theta_1 x_{11} + \theta_2 x_{12} \\ \theta_0 + \theta_1 x_{21} + \theta_2 x_{22} \\ \vdots \\ \theta_0 + \theta_1 x_{n1} + \theta_2 x_{n2} \end{pmatrix} = \hat{y}_i = \theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} = \hat{Y}$$

d) $SSE = (Y-\hat{Y})^T(Y-\hat{Y})$

$= (Y-X\Theta)^T(Y-X\Theta)$

$= Y^TY - \Theta^TX^TY - \Theta^TX^TY + \Theta^TX^TX\Theta$

$= Y^TY - 2Y^TX\Theta + \Theta^TX^TX\Theta$

$\frac{\partial SSE}{\partial \Theta} = \frac{\partial}{\partial \Theta}(Y^TY) - \frac{\partial}{\partial \Theta}(2Y^TX\Theta) + \frac{\partial}{\partial \Theta}(\Theta^TX^TX\Theta)$

$= 0 - 2X^TY + 2(X^TX)\Theta$

e) $0 = -2X^TY + 2(X^TX)\Theta$

$X^TY = (X^TX)\Theta$

$(X^TX)^{-1}X^TY = \Theta$

Question 5

Part A

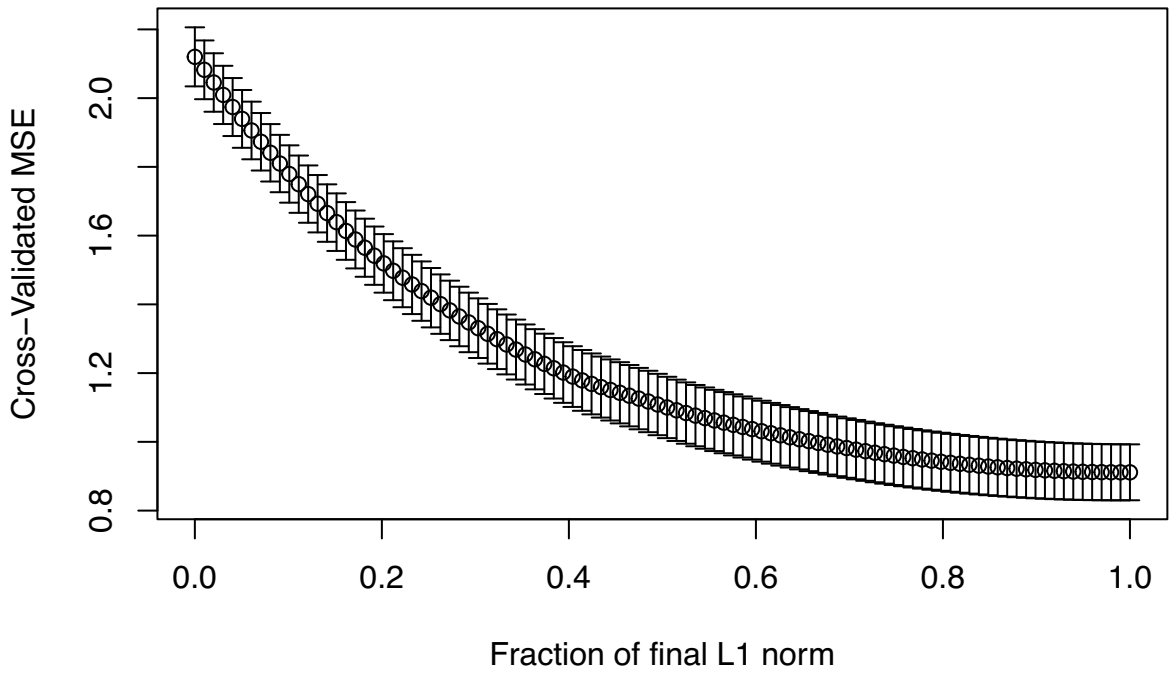
```
lasso <-lars(x=as.matrix(fishToxicity[,1:6]),y=as.numeric(unlist(fishToxicity[,7])),trace=TRUE)
```

```
## LASSO sequence
## Computing X'X .....
## LARS Step 1 :     Variable 6     added
## LARS Step 2 :     Variable 2     added
## LARS Step 3 :     Variable 3     added
## LARS Step 4 :     Variable 4     added
## LARS Step 5 :     Variable 5     added
## LARS Step 6 :     Variable 1     added
## Computing residuals, RSS etc .....
```

```
#Based on the Output the following models will be listed based on the LASSO Method
# 1st Model: X_6 (MLOGP)
# 2nd Model: X_6 (MLOGP), X_2 (SM1_Dx(Z))
# 3rd Model: X_6 (MLOGP), X_2 (SM1_Dx(Z)), X_3(GATS1i)
# 4th Model: X_6 (MLOGP), X_2 (SM1_Dx(Z)), X_3(GATS1i), X_4(NdsCH)
# 5th Model: X_6 (MLOGP), X_2 (SM1_Dx(Z)), X_3(GATS1i), X_4(NdsCH), X_5 (NdssC)
# 6th Model: X_6 (MLOGP), X_2 (SM1_Dx(Z)), X_3(GATS1i), X_4(NdsCH), X_5 (NdssC), X_1 (CIC0)
```

Part B

```
cv.lars(x=as.matrix(fishToxicity[,1:6]),y=as.numeric(unlist(fishToxicity[,7])),K=10)
```




```
# Based on looking at the plot, I would choose 0.8 as the best fraction s value.
```

Part C

```
lasso <-lars(x=as.matrix(fishToxicity[,1:6]),y=as.numeric(unlist(fishToxicity[,7])),trace=TRUE)
```

```
## LASSO sequence
## Computing X'X .....
## LARS Step 1 :     Variable 6     added
## LARS Step 2 :     Variable 2     added
## LARS Step 3 :     Variable 3     added
## LARS Step 4 :     Variable 4     added
## LARS Step 5 :     Variable 5     added
## LARS Step 6 :     Variable 1     added
## Computing residuals, RSS etc .....
```

```
coef(lasso,s=0.8,mode="fraction")
```

```
##       CIC0  SM1_Dz.Z.     GATS1i      NdsCH      NdssC      MLOGP
##  0.2077567  0.9921246 -0.4651972  0.2890276  0.0354972  0.4324455
```

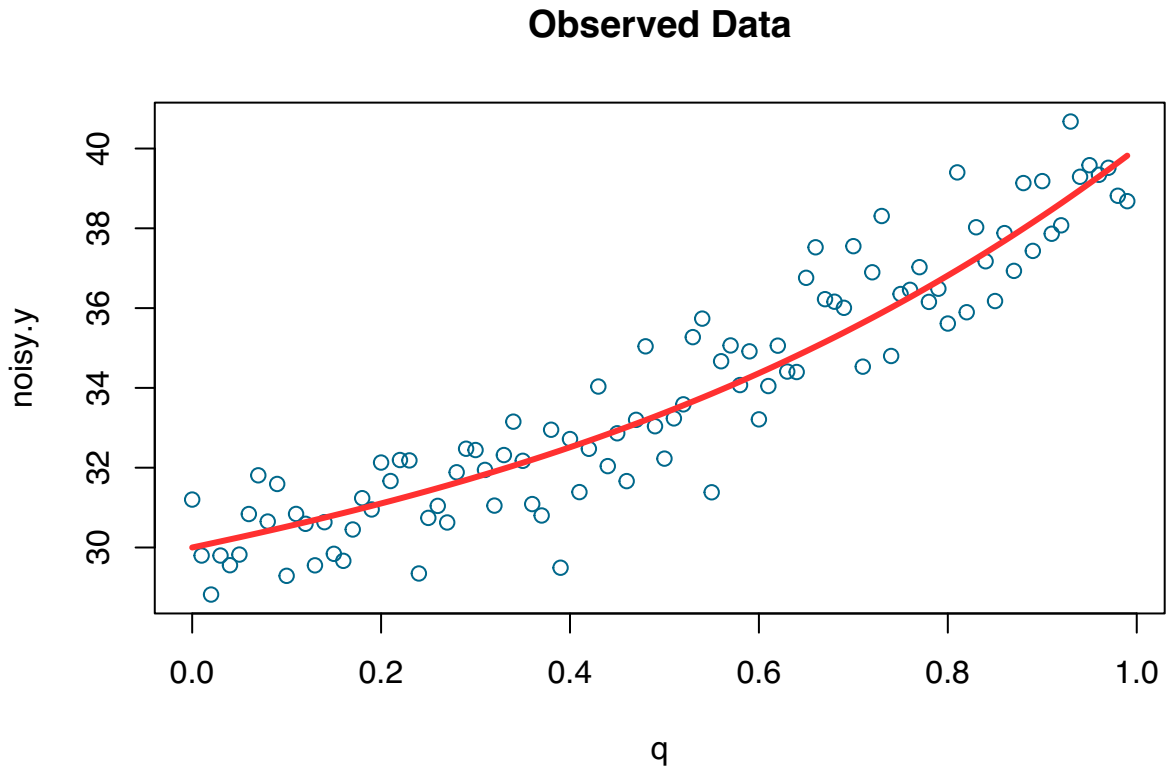
Question 6

Part A

```
q = seq(0, .99, by = 0.01)
length(q)
```

```
## [1] 100
```

```
y = 30 + 5*q + 2*q^2 + 3*q^3
noise = rnorm(length(q), mean = 0, sd = 1)
noisy.y = y + noise
plot(q, noisy.y, col = 'deepskyblue4', xlab = 'q', main = 'Observed Data')
lines(q, y, col = 'firebrick1', lwd = 3)
```




```
model <- lm(noisy.y ~ q + I(q^2) + I(q^3))
summary(model)
```

```
##
## Call:
## lm(formula = noisy.y ~ q + I(q^2) + I(q^3))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.84753 -0.71749  0.02206  0.71125  2.15023
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  30.2033     0.4075  74.115   <2e-16 ***
## q             0.2893     3.5829   0.081   0.9358
## I(q^2)       16.0228     8.4337   1.900   0.0605 .
```

```
## I(q^3)       -6.9555     5.5983  -1.242   0.2171
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.057 on 96 degrees of freedom
## Multiple R-squared:  0.8882, Adjusted R-squared:  0.8847
## F-statistic: 254.2 on 3 and 96 DF,  p-value: < 2.2e-16
```

```
# We get the model to be 29.9287 + 0.5647x + 16.7805x^2 - 7.5869x^3
#which is quite different from the true model

#We get the R squared for the estimated model to be 0.8886
```

Part B

```
matdata = c(1, q[1], (q[1]^2), (q[1]^3), 1, q[2], (q[2]^2), (q[2]^3))
xMat = matrix(matdata, nrow = 2, ncol = 4, byrow = TRUE)
colnames(xMat) = c('1', 'x', 'x^2', 'x^3')
rownames(xMat) = c('row 1', 'row 2')
xMat
```

```
##       1    x   x^2   x^3
## row 1 1 0.00 0e+00 0e+00
## row 2 1 0.01 1e-04 1e-06
```

Part C

```
model2 = lm(noisy.y ~ q + I(q^2) + I(q^3) + I(q^4))
summary(model2)
```

```
##
## Call:
## lm(formula = noisy.y ~ q + I(q^2) + I(q^3) + I(q^4))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.81946 -0.69717  0.00276  0.71388  2.15258
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  30.0869     0.5006  60.100   <2e-16 ***
## q             2.7527     7.0816   0.389    0.698
## I(q^2)        4.6941    29.3004   0.160    0.873
## I(q^3)       10.9040    44.5754   0.245    0.807
## I(q^4)       -9.0200    22.3330  -0.404    0.687
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.062 on 95 degrees of freedom
## Multiple R-squared:  0.8884, Adjusted R-squared:  0.8837
## F-statistic:   189 on 4 and 95 DF,  p-value: < 2.2e-16
```

```
# We get the model to be 30.324 - 7.806x + 55.279x^2 - 68.280x^3 + 30.653x^4
#Once again, this is quite different from the true model in-terms of the coefficients
```

```
# We get the R squared for the estimated model to be 0.8887.
#It has increased by a little, due to the fact that we have added one more regressor variable.
#The more regressors we had, the bigger the R squared tends to 1.
```