Homework 2

Author: Dong Bin

email: bindong2@illinois.edu

Question 1

Part a

```
rm(list = ls())
library(mlbench)
data(BostonHousing2)
BH = BostonHousing2[, !(colnames(BostonHousing2) %in% c("medv", "town", "tract"))]

# Get some basic informations
dim(BH)
```

```
## [1] 506  16
```

```
names(BH)
```

```
##  [1] "lon"     "lat"     "cmedv"   "crim"    "zn"      "indus"   "chas"
##  [8] "nox"     "rm"      "age"     "dis"     "rad"     "tax"     "ptratio"
## [15] "b"       "lstat"
```

```
# Fit a LM model
full.model <- lm(cmedv~., data = BH)
summary(full.model)
```

```
##
## Call:
## lm(formula = cmedv ~ ., data = BH)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.5831  -2.7643  -0.5994   1.7482  26.0822
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.350e+02  3.032e+02  -1.435 0.152029
## lon         -3.935e+00  3.372e+00  -1.167 0.243770
## lat          4.495e+00  3.669e+00   1.225 0.221055
## crim        -1.045e-01  3.261e-02  -3.206 0.001436 **
## zn           4.657e-02  1.374e-02   3.390 0.000755 ***
## indus        1.524e-02  6.175e-02   0.247 0.805106
## chas1        2.578e+00  8.650e-01   2.980 0.003024 **
## nox         -1.582e+01  4.005e+00  -3.951 8.93e-05 ***
## rm           3.754e+00  4.166e-01   9.011  < 2e-16 ***
```

```
## age          2.468e-03  1.335e-02   0.185 0.853440
## dis         -1.400e+00  2.088e-01  -6.704 5.61e-11 ***
## rad          3.067e-01  6.658e-02   4.607 5.23e-06 ***
## tax         -1.289e-02  3.727e-03  -3.458 0.000592 ***
## ptratio     -8.771e-01  1.363e-01  -6.436 2.92e-10 ***
## b            9.176e-03  2.663e-03   3.446 0.000618 ***
## lstat       -5.374e-01  5.042e-02 -10.660  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.7 on 490 degrees of freedom
## Multiple R-squared:  0.7458, Adjusted R-squared:  0.738
## F-statistic: 95.82 on 15 and 490 DF,  p-value: < 2.2e-16
```

The most significant variables according to P value are: 1. rm

2. lstat

Part b

```
p <- dim(BH)[2]

test <- step(full.model, k = log(p))
```

```
## Start:  AIC=1594.28
## cmedv ~ lon + lat + crim + zn + indus + chas + nox + rm + age +
##     dis + rad + tax + ptratio + b + lstat
##
##           Df Sum of Sq   RSS    AIC
## - age      1      0.75 10826 1591.5
## - indus    1      1.35 10826 1591.6
## - lon      1     30.09 10855 1592.9
## - lat      1     33.17 10858 1593.0
## <none>                 10825 1594.3
## - chas     1    196.21 11021 1600.6
## - crim     1    227.01 11052 1602.0
## - zn       1    253.89 11079 1603.2
## - b        1    262.35 11087 1603.6
## - tax      1    264.16 11089 1603.7
## - nox      1    344.85 11170 1607.4
## - rad      1    468.79 11294 1613.0
## - ptratio  1    915.13 11740 1632.6
## - dis      1    992.75 11818 1635.9
## - rm       1   1793.75 12619 1669.1
## - lstat    1   2510.61 13336 1697.0
##
## Step:  AIC=1591.54
## cmedv ~ lon + lat + crim + zn + indus + chas + nox + rm + dis +
##     rad + tax + ptratio + b + lstat
##
##           Df Sum of Sq   RSS    AIC
```

```
## - indus    1      1.42 10827 1588.8
## - lon      1     29.36 10855 1590.1
## - lat      1     33.69 10859 1590.3
## <none>                 10826 1591.5
## - chas     1    199.53 11025 1598.0
## - crim     1    227.26 11053 1599.3
## - zn       1    253.44 11079 1600.5
## - tax      1    263.54 11089 1600.9
## - b        1    264.77 11090 1601.0
## - nox      1    352.01 11178 1605.0
## - rad      1    468.06 11294 1610.2
## - ptratio  1    914.57 11740 1629.8
## - dis      1   1122.29 11948 1638.7
## - rm       1   1905.55 12731 1670.8
## - lstat    1   2804.14 13630 1705.3
##
## Step:  AIC=1588.83
## cmedv ~ lon + lat + crim + zn + chas + nox + rm + dis + rad +
##     tax + ptratio + b + lstat
##
##           Df Sum of Sq   RSS    AIC
## - lon      1     32.13 10859 1587.6
## - lat      1     33.34 10860 1587.6
## <none>                 10827 1588.8
## - chas     1    203.29 11030 1595.5
## - crim     1    228.39 11056 1596.6
## - zn       1    252.83 11080 1597.7
## - b        1    263.94 11091 1598.2
## - tax      1    303.69 11131 1600.1
## - nox      1    372.59 11200 1603.2
## - rad      1    495.77 11323 1608.7
## - ptratio  1    929.67 11757 1627.7
## - dis      1   1173.09 12000 1638.1
## - rm       1   1915.38 12742 1668.5
## - lstat    1   2813.97 13641 1703.0
##
## Step:  AIC=1587.56
## cmedv ~ lat + crim + zn + chas + nox + rm + dis + rad + tax +
##     ptratio + b + lstat
##
##           Df Sum of Sq   RSS    AIC
## - lat      1     24.92 10884 1586.0
## <none>                 10859 1587.6
## - chas     1    235.62 11095 1595.7
## - crim     1    240.30 11100 1595.9
## - b        1    258.02 11117 1596.7
## - zn       1    281.72 11141 1597.8
## - tax      1    288.66 11148 1598.1
## - nox      1    504.45 11364 1607.8
## - rad      1    511.62 11371 1608.1
## - ptratio  1   1137.60 11997 1635.2
## - dis      1   1406.62 12266 1646.4
## - rm       1   1946.89 12806 1668.2
## - lstat    1   2810.02 13669 1701.2
```

```
##
## Step:  AIC=1585.95
## cmedv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio +
##     b + lstat
##
##           Df Sum of Sq   RSS    AIC
## <none>                 10884 1586.0
## - chas     1    228.64 11113 1593.7
## - crim     1    237.49 11122 1594.1
## - b        1    265.68 11150 1595.4
## - zn       1    272.12 11156 1595.7
## - tax      1    287.68 11172 1596.4
## - rad      1    490.76 11375 1605.5
## - nox      1    538.23 11422 1607.6
## - ptratio  1   1132.44 12017 1633.3
## - dis      1   1502.93 12387 1648.6
## - rm       1   1940.06 12824 1666.2
## - lstat    1   2785.20 13669 1698.5
```

The variables removed from full model after stepwise regression with BIC criteria are: 1. age

2. indus
3. lon
4. lat

Part c

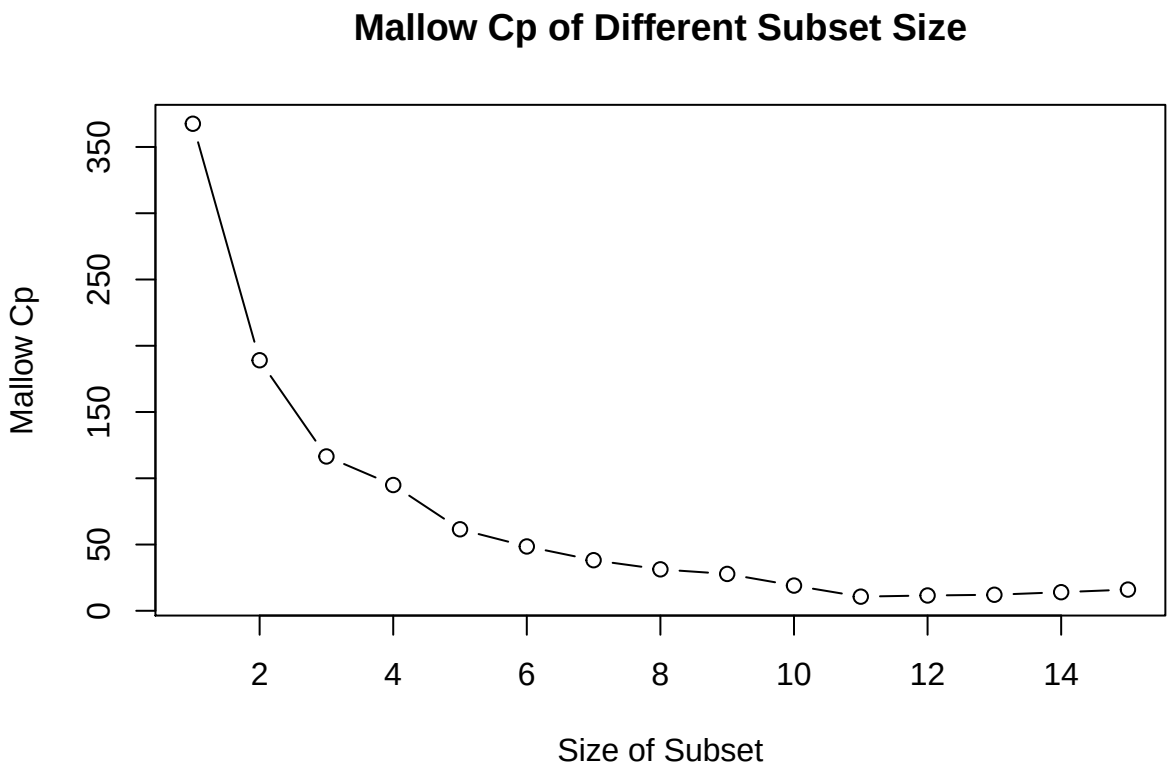
```
library(leaps)
b = regsubsets(cmedv~ ., data = BH, nvmax =p)
rs = summary(b)
rs$which
```

```
##    (Intercept)   lon   lat  crim    zn indus chas1   nox    rm   age   dis
## 1         TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 2         TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
## 3         TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
## 4         TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE
## 5         TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE
## 6         TRUE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE
## 7         TRUE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE
## 8         TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE
## 9         TRUE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE
## 10        TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE
## 11        TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE
## 12        TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE
## 13        TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE
## 14        TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
## 15        TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##      rad   tax ptratio     b lstat
## 1  FALSE FALSE   FALSE FALSE  TRUE
```

```
## 2  FALSE FALSE   FALSE FALSE  TRUE
## 3  FALSE FALSE    TRUE FALSE  TRUE
## 4  FALSE FALSE    TRUE FALSE  TRUE
## 5  FALSE FALSE    TRUE FALSE  TRUE
## 6  FALSE FALSE    TRUE FALSE  TRUE
## 7  FALSE FALSE    TRUE  TRUE  TRUE
## 8  FALSE FALSE    TRUE  TRUE  TRUE
## 9   TRUE  TRUE    TRUE  TRUE  TRUE
## 10  TRUE  TRUE    TRUE  TRUE  TRUE
## 11  TRUE  TRUE    TRUE  TRUE  TRUE
## 12  TRUE  TRUE    TRUE  TRUE  TRUE
## 13  TRUE  TRUE    TRUE  TRUE  TRUE
## 14  TRUE  TRUE    TRUE  TRUE  TRUE
## 15  TRUE  TRUE    TRUE  TRUE  TRUE
```

Part d

```
row <- rs$which[1,]
names <- names(BH)
xlabel <- c(1:15)
plot(x = xlabel, y = rs$cp, type="b", main="Mallow Cp of Different Subset Size", xlab = "Size of Subset
```



```
rs$which[11,]
```

```
## (Intercept)         lon         lat        crim          zn       indus
##        TRUE       FALSE       FALSE        TRUE        TRUE       FALSE
##       chas1         nox          rm         age         dis         rad
##        TRUE        TRUE        TRUE       FALSE        TRUE        TRUE
##         tax     ptratio           b       lstat
##        TRUE        TRUE        TRUE        TRUE
```

The best model is when model size is 11. Remaining variables are: crim, zn, chas1, nox, rm, dis, rad, tax, ptratio, b, lstat

```
SubData <- BostonHousing2[, (colnames(BostonHousing2) %in% c("cmedv","crim", "zn", "chas1", "nox", "rm"
head(SubData)
```

```
##   cmedv    crim zn   nox    rm    dis rad tax ptratio      b lstat
## 1  24.0 0.00632 18 0.538 6.575 4.0900   1 296    15.3 396.90  4.98
## 2  21.6 0.02731  0 0.469 6.421 4.9671   2 242    17.8 396.90  9.14
## 3  34.7 0.02729  0 0.469 7.185 4.9671   2 242    17.8 392.83  4.03
## 4  33.4 0.03237  0 0.458 6.998 6.0622   3 222    18.7 394.63  2.94
## 5  36.2 0.06905  0 0.458 7.147 6.0622   3 222    18.7 396.90  5.33
## 6  28.7 0.02985  0 0.458 6.430 6.0622   3 222    18.7 394.12  5.21
```

```
summary(lm(cmedv~., data=SubData))
```

```
##
## Call:
## lm(formula = cmedv ~ ., data = SubData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.3325  -2.7562  -0.5958   1.9273  26.3651
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.524865   5.068796   7.206 2.17e-12 ***
## crim         -0.112317   0.032745  -3.430 0.000654 ***
## zn            0.046996   0.013528   3.474 0.000558 ***
## nox         -16.407119   3.525175  -4.654 4.18e-06 ***
## rm            3.821857   0.406256   9.408  < 2e-16 ***
## dis          -1.553761   0.185510  -8.376 5.70e-16 ***
## rad           0.312528   0.063230   4.943 1.06e-06 ***
## tax          -0.012976   0.003362  -3.860 0.000129 ***
## ptratio      -0.949052   0.128728  -7.373 7.09e-13 ***
## b             0.009643   0.002671   3.610 0.000338 ***
## lstat        -0.534008   0.047412 -11.263  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.738 on 495 degrees of freedom
## Multiple R-squared:  0.739,  Adjusted R-squared:  0.7337
## F-statistic: 140.2 on 10 and 495 DF,  p-value: < 2.2e-16
```

After removing insignificant variables, the most significant variables are :

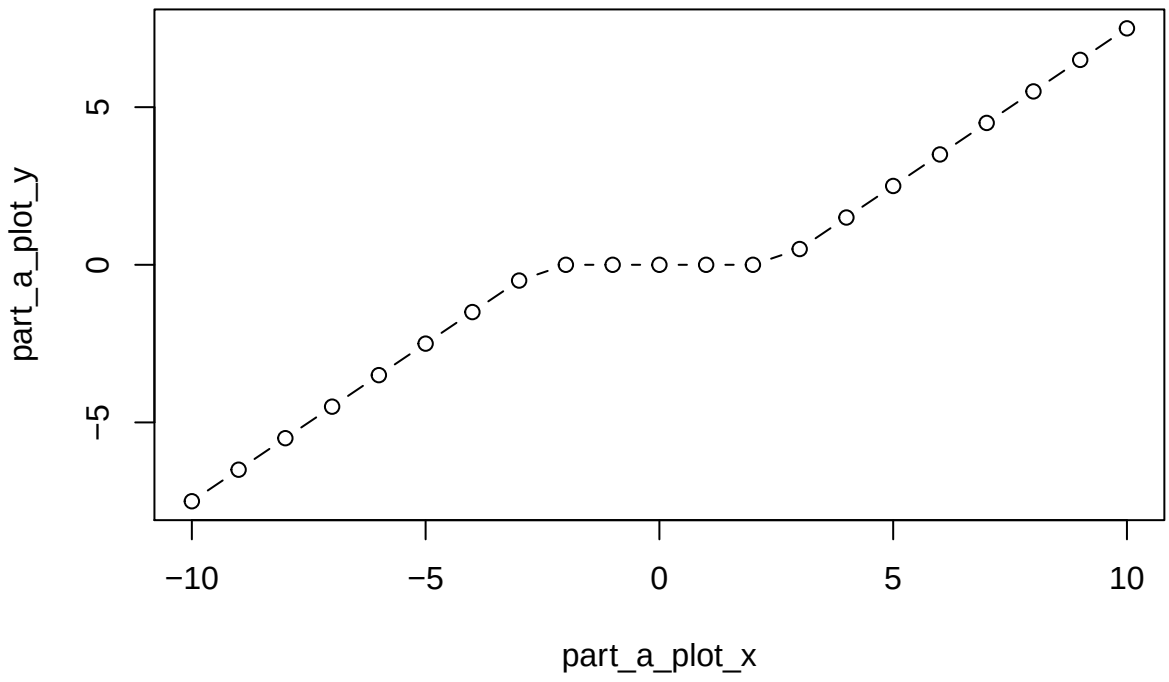
1. rm
2. lstat

Question 2

```
rm(list=ls())
library(MASS)
set.seed(1)
n <- 200
p <- 200

# generate data
V <- matrix(0.2, p, p)
diag(V) <- 1
X <- as.matrix(mvrnorm(n, mu = rep(0, p), Sigma = V))
y <- X[, 1] + 0.5*X[, 2] + 0.25*X[, 3] + rnorm(n)

# we will use a scaled version
X <- scale(X)
y <- scale(y)
```

Part a

```
soft_th <- function(b, lambda){
  sign(b) * max(abs(b)-lambda/2, 0);
}

part_a_plot_x <- c(-10:10)
part_a_plot_y <- list()
for(i in c(1:length(part_a_plot_x)))
{
  part_a_plot_y[i] <- soft_th(part_a_plot_x[i], 5)
}
plot(part_a_plot_x, part_a_plot_y, type <- "b")
```



Part b

```
beta_old <- rep(0, p)
update_beta <- function(beta, lambda, X, y, printable)
{
  current_beta <- beta;
  r <- y-X%*%beta
  for(i in 1:length(current_beta))
  {
    r = r+ X[,i]*current_beta[i]
    current_beta[i] = soft_th(X[,i]%*%r, lambda*ncol(X))/(t(X[,i]%*%X[,i]))
    r = r-X[,i]*current_beta[i]
    r <- y-X%*%current_beta
    if(i<=3 && printable)
    {
      cat("Obervation of r: ",  i, "\n")
      print(c(r))
    }
  }
  return <-current_beta
}
a = update_beta(beta_old, 0.7, X, y, TRUE)
```

```
## Obervation of r:  1
```

```
##   [1] -0.1732248160  0.0420159502  0.1017413893 -0.7190409534  0.8735648528
##   [6]  0.3043390600  1.5503897913  0.5914017304 -0.2693232724 -0.7999729342
##  [11] -0.6818717787  0.5079830534 -1.1633191170  0.9688017899 -0.9794257473
##  [16]  0.4225879617  1.1081421403 -0.3156059563 -1.0970589191 -0.3271211892
##  [21] -0.4622883491 -1.3908223009 -1.3295801037 -0.3894549681 -0.0255554232
##  [26] -1.4118107691  0.1894732829  2.1611294225 -0.0298275861  0.0852998151
##  [31] -0.4459571604  1.9615653392 -0.7954281586 -0.3092378168  0.7059809402
##  [36]  0.4629458583 -1.1288951338  0.0884962406 -0.0532806694 -0.5047273356
##  [41] -0.5000061093  1.0205571751  0.4867703214 -0.5852535770  0.6052498628
##  [46] -0.0879819949 -0.6241886988  0.4138138991  0.1171855528 -0.7370632183
##  [51]  0.1799611854  0.2395890119 -0.4353521252  0.4071855453 -0.4541504673
##  [56] -0.9260570476 -0.3947275004  0.0052101059 -1.1888943738 -0.2497149380
##  [61] -1.6025001712 -1.1162071692 -1.0214247291  0.0753905095  0.3556351120
##  [66] -0.9642238275  1.0210046942 -1.2718623800  0.0081743843  0.5359956784
##  [71]  0.4218509015  0.3375661249 -0.1486921898  0.5237373709  0.3663538101
##  [76]  0.6213726132 -0.3053890679 -0.7394843237 -0.5884643249  2.1275937180
##  [81] -0.2371954547  1.9505948639 -0.3501185135  2.0307462747  1.3369680729
##  [86]  0.1822922785 -0.3135356767 -0.3724899490  0.2704827710  0.2289949076
##  [91] -0.2402292780 -0.6305157116 -0.8964669069  0.5605898657 -0.6921811218
##  [96] -0.3289906484 -0.4621565795 -0.3765847464  1.1890883772  1.0437079605
## [101]  0.9637565479  0.5220322942 -0.8316114016  0.8346767395 -0.9354648061
## [106] -0.7975758609 -2.0326058116 -0.5693907224 -0.3015305706 -0.7411136048
## [111] -0.4068442225  0.1812417606 -0.2956630841 -0.6439320317 -0.5077425522
## [116]  0.1244277272 -0.3106782657 -0.1914507410 -0.0502843569 -0.3713660103
## [121]  0.4901625161  0.7798041635 -0.5761134297  0.7480087944 -0.5730341085
## [126]  1.3249910360  1.1485287261 -1.1858135230  0.5575931757 -0.8449040174
## [131]  0.0336300022 -0.8368536471  0.5899360596  0.3172054440 -1.0196181899
## [136]  1.5690933417  0.2155800925 -0.6798071745  0.0255887080  0.8911107790
## [141]  0.1710874556 -0.0005850383  0.5204731950  1.0932089437 -0.4595836670
## [146]  0.8386680889 -1.3484513128  1.2933431646  0.7085560111  1.0601727115
## [151] -0.2759870560 -0.1484081333 -0.4195945340  0.6412570205  1.3592933892
## [156] -0.0392419103 -0.5925756281  0.5070265016  0.7187171660  0.0189690173
## [161] -0.2616882198  0.3675901004  0.1481630956 -0.2335605960  0.9909879885
## [166] -1.5486344501 -1.2154010826  0.4529784058 -0.3570059117  0.2200198505
## [171] -0.4959727414  0.3518812356  0.4016313396 -0.5245097039  0.7864117757
## [176] -0.3905440521 -1.5176221661 -1.4992827151  0.2436098174 -0.1487278477
## [181]  0.9374834744 -0.1143371495  0.1477487198  0.8977124799 -1.2726361333
## [186] -0.2316854754 -0.6218504219  0.2333298967  0.1662430937  0.1936489031
## [191] -0.2527393941 -0.7543181277  1.2890207543  0.4996340247  1.7384160367
## [196]  0.1710833742  0.2206195699  0.1661604430  1.4315116331 -0.0202732725
## Obervation of r:  2
##   [1] -0.0778719330  0.1462448282  0.1576183753 -0.6386927853  0.7017861214
##   [6]  0.2821582791  1.4642109899  0.5099568920 -0.3129919583 -0.8448934262
##  [11] -0.5128195822  0.5410094658 -1.1522049299  0.7988161370 -0.8473630701
##  [16]  0.4982688430  1.0231786398 -0.3863871229 -1.0444752639 -0.4165628355
##  [21] -0.4667624777 -1.3698527994 -1.2959965039 -0.4066916399 -0.0425985044
##  [26] -1.2934998490  0.1152721273  1.9861322497  0.0029177384  0.2293051444
##  [31] -0.4911115640  1.8984550406 -0.7827626900 -0.2916580657  0.8410866075
##  [36]  0.4835737464 -1.0178453619  0.0585766635  0.0216586169 -0.4839968249
##  [41] -0.5472627749  0.8722722727  0.5294123881 -0.5473446819  0.4285696705
##  [46] -0.1879268690 -0.5648223828  0.4034933230  0.1093268156 -0.6008592051
##  [51]  0.3297371627  0.2358271006 -0.5382788508  0.3441676851 -0.3521256832
##  [56] -0.8458061261 -0.4901636105  0.0115083314 -1.0886150247 -0.2412133362
##  [61] -1.4522037330 -1.0212441809 -1.0349841091  0.0372360149  0.4294521283
```

```
##  [66] -0.8093210875  0.9074852400 -1.2230477457 -0.0489291419  0.5795263060
##  [71]  0.3911637377  0.3840320595 -0.1741223400  0.4442190361  0.3356239129
##  [76]  0.6491041807 -0.3176856600 -0.6724241362 -0.6374801176  2.1247959783
##  [81] -0.2943092646  1.8537011916 -0.2244468875  1.9639627285  1.3226473048
##  [86]  0.1400726412 -0.2308983629 -0.3826003534  0.2170102423  0.2649676393
##  [91] -0.2154780388 -0.5575712984 -0.8283400153  0.4767692474 -0.4903652468
##  [96] -0.2712797042 -0.3949519729 -0.2678311872  1.1205455105  0.8903435411
## [101]  0.9525434606  0.5162440040 -0.6045225009  0.6940899894 -0.9334806114
## [106] -0.6989701586 -1.8686414392 -0.3619375899 -0.2935372169 -0.7071951164
## [111] -0.4715281004  0.1565633260 -0.2847683710 -0.6490492431 -0.3752062803
## [116]  0.1350501976 -0.3193632457 -0.2514927429 -0.0172811679 -0.2339102270
## [121]  0.5598243231  0.9180845951 -0.4858398739  0.5950353115 -0.4786899906
## [126]  1.3115365170  1.0288636274 -1.2623712707  0.5205101608 -0.8675494406
## [131] -0.0432292995 -0.7779365753  0.4792063172  0.3468894294 -1.0225911616
## [136]  1.4492951597  0.1648482809 -0.5941595793  0.0369103809  0.8915186479
## [141]  0.1059075555 -0.0074025370  0.4027428819  0.9498073395 -0.5744628720
## [146]  0.7123297104 -1.1322879514  1.1739602324  0.6973263708  1.0385793580
## [151] -0.1935488588 -0.1797454263 -0.3440201579  0.5377822341  1.3430360463
## [156] -0.0873464917 -0.4834086398  0.4755979426  0.6708799822  0.0199035957
## [161] -0.2662991033  0.4122704199  0.1096018455 -0.2340020697  0.8327312175
## [166] -1.5274413836 -1.0937778201  0.4765903117 -0.3621414621  0.1024878310
## [171] -0.3950573435  0.2775350167  0.3712493619 -0.6602079700  0.6188914627
## [176] -0.4389077768 -1.4486895886 -1.4626021549  0.2801111409  0.0008485366
## [181]  0.9403161831 -0.0029126007  0.0719572172  0.7803684134 -1.1278227144
## [186] -0.3576253425 -0.6360190253  0.1436849005  0.0478617338  0.1283461044
## [191] -0.2509829528 -0.7143911147  1.2959706914  0.5867198404  1.5088009226
## [196]  0.1294935267  0.3680433780  0.0768735710  1.3182956889 -0.0057893126
## Obervation of r:  3
##   [1] -0.0778719330  0.1462448282  0.1576183753 -0.6386927853  0.7017861214
##   [6]  0.2821582791  1.4642109899  0.5099568920 -0.3129919583 -0.8448934262
##  [11] -0.5128195822  0.5410094658 -1.1522049299  0.7988161370 -0.8473630701
##  [16]  0.4982688430  1.0231786398 -0.3863871229 -1.0444752639 -0.4165628355
##  [21] -0.4667624777 -1.3698527994 -1.2959965039 -0.4066916399 -0.0425985044
##  [26] -1.2934998490  0.1152721273  1.9861322497  0.0029177384  0.2293051444
##  [31] -0.4911115640  1.8984550406 -0.7827626900 -0.2916580657  0.8410866075
##  [36]  0.4835737464 -1.0178453619  0.0585766635  0.0216586169 -0.4839968249
##  [41] -0.5472627749  0.8722722727  0.5294123881 -0.5473446819  0.4285696705
##  [46] -0.1879268690 -0.5648223828  0.4034933230  0.1093268156 -0.6008592051
##  [51]  0.3297371627  0.2358271006 -0.5382788508  0.3441676851 -0.3521256832
##  [56] -0.8458061261 -0.4901636105  0.0115083314 -1.0886150247 -0.2412133362
##  [61] -1.4522037330 -1.0212441809 -1.0349841091  0.0372360149  0.4294521283
##  [66] -0.8093210875  0.9074852400 -1.2230477457 -0.0489291419  0.5795263060
##  [71]  0.3911637377  0.3840320595 -0.1741223400  0.4442190361  0.3356239129
##  [76]  0.6491041807 -0.3176856600 -0.6724241362 -0.6374801176  2.1247959783
##  [81] -0.2943092646  1.8537011916 -0.2244468875  1.9639627285  1.3226473048
##  [86]  0.1400726412 -0.2308983629 -0.3826003534  0.2170102423  0.2649676393
##  [91] -0.2154780388 -0.5575712984 -0.8283400153  0.4767692474 -0.4903652468
##  [96] -0.2712797042 -0.3949519729 -0.2678311872  1.1205455105  0.8903435411
## [101]  0.9525434606  0.5162440040 -0.6045225009  0.6940899894 -0.9334806114
## [106] -0.6989701586 -1.8686414392 -0.3619375899 -0.2935372169 -0.7071951164
## [111] -0.4715281004  0.1565633260 -0.2847683710 -0.6490492431 -0.3752062803
## [116]  0.1350501976 -0.3193632457 -0.2514927429 -0.0172811679 -0.2339102270
## [121]  0.5598243231  0.9180845951 -0.4858398739  0.5950353115 -0.4786899906
## [126]  1.3115365170  1.0288636274 -1.2623712707  0.5205101608 -0.8675494406
```

```
## [131] -0.0432292995 -0.7779365753  0.4792063172  0.3468894294 -1.0225911616
## [136]  1.4492951597  0.1648482809 -0.5941595793  0.0369103809  0.8915186479
## [141]  0.1059075555 -0.0074025370  0.4027428819  0.9498073395 -0.5744628720
## [146]  0.7123297104 -1.1322879514  1.1739602324  0.6973263708  1.0385793580
## [151] -0.1935488588 -0.1797454263 -0.3440201579  0.5377822341  1.3430360463
## [156] -0.0873464917 -0.4834086398  0.4755979426  0.6708799822  0.0199035957
## [161] -0.2662991033  0.4122704199  0.1096018455 -0.2340020697  0.8327312175
## [166] -1.5274413836 -1.0937778201  0.4765903117 -0.3621414621  0.1024878310
## [171] -0.3950573435  0.2775350167  0.3712493619 -0.6602079700  0.6188914627
## [176] -0.4389077768 -1.4486895886 -1.4626021549  0.2801111409  0.0008485366
## [181]  0.9403161831 -0.0029126007  0.0719572172  0.7803684134 -1.1278227144
## [186] -0.3576253425 -0.6360190253  0.1436849005  0.0478617338  0.1283461044
## [191] -0.2509829528 -0.7143911147  1.2959706914  0.5867198404  1.5088009226
## [196]  0.1294935267  0.3680433780  0.0768735710  1.3182956889 -0.0057893126
```

```
cat("Value of beta after one loop: \n")
```

```
## Value of beta after one loop:
```

```
print(a)
```

```
##   [1] 0.35120464 0.08904678 0.00000000 0.00000000 0.00000000 0.00000000
##   [7] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##  [13] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##  [19] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##  [25] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##  [31] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##  [37] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##  [43] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##  [49] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##  [55] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##  [61] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##  [67] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##  [73] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##  [79] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##  [85] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##  [91] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##  [97] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## [103] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## [109] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## [115] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## [121] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## [127] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## [133] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## [139] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## [145] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## [151] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## [157] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## [163] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## [169] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## [175] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## [181] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## [187] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
```

```
## [193] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## [199] 0.00000000 0.00000000
```

Part c

```
myLasso <- function(X, y, lambda, tol, maxitr)
{
  old_beta = rep(0, p);
  for( i in 1:maxitr)
  {
    cat("Iteration: ", i, "\n")
    new_beta = update_beta(old_beta, lambda, X, y, FALSE);
    residule = sum(abs(new_beta-old_beta))
    if(i<=3)
    {
      print(residule)
    }
    if(residule < tol){
      cat("Terminated after ", i, " ietrations", "\n")
      break;
    }


    old_beta = new_beta;
  }
  return <- old_beta
}

myfit <- myLasso(X, y, 0.3, 1e-5, 100)
```

```
## Iteration:  1
## [1] 0.8882459
## Iteration:  2
## [1] 0.1139365
## Iteration:  3
## [1] 0.01282938
## Iteration:  4
## Iteration:  5
## Iteration:  6
## Iteration:  7
## Iteration:  8
## Iteration:  9
## Terminated after  9  ietrations
```

```
for(i in 1:length(myfit))
{
  if(myfit[i]!=0)
  {
    cat(i, myfit[i], "\n");
  }
}
```

```
## 1 0.4574072
## 2 0.2257711
## 3 0.1141653
## 14 0.0005767683
## 118 0.01112198
## 137 0.004317203
```

Part d

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-18
```

```
lasso = glmnet(X, y, alpha=1, lambda=0.15)
summary(lasso$beta)
```

```
## 200 x 1 sparse Matrix of class "dgCMatrix", with 6 entries
##     i j            x
## 1   1 1 0.4576113537
## 2   2 1 0.2259454383
## 3   3 1 0.1142807376
## 4  14 1 0.0007967669
## 5 118 1 0.0113352128
## 6 137 1 0.0044930898
```

```
distance <- sum(abs(myfit-lasso$beta))
cat("The distance between my implementation and glmnet is: ", distance, "\n")
```

```
## The distance between my implementation and glmnet is:  0.001103005
```

Question 3

Part a

```
rm(list=ls())
# Read data into R
data <- read.csv('Train.csv', header=TRUE)

str(data)
```

```
## 'data.frame':    8523 obs. of  12 variables:
##  $ Item_Identifier          : Factor w/ 1559 levels "DRA12","DRA24",..: 157 9 663 1122 1298 759 697 
##  $ Item_Weight              : num  9.3 5.92 17.5 19.2 8.93 ...
##  $ Item_Fat_Content         : Factor w/ 5 levels "LF","low fat",..: 3 5 3 5 3 5 5 3 5 5 ...
##  $ Item_Visibility          : num  0.016 0.0193 0.0168 0 0 ...
##  $ Item_Type                : Factor w/ 16 levels "Baking Goods",..: 5 15 11 7 10 1 14 14 6 6 ...
##  $ Item_MRP                 : num  249.8 48.3 141.6 182.1 53.9 ...
##  $ Outlet_Identifier        : Factor w/ 10 levels "OUT010","OUT013",..: 10 4 10 1 2 4 2 6 8 3 ...
##  $ Outlet_Establishment_Year: int  1999 2009 1999 1998 1987 2009 1987 1985 2002 2007 ...
##  $ Outlet_Size              : Factor w/ 4 levels "","High","Medium",..: 3 3 3 1 2 3 2 3 1 1 ...
##  $ Outlet_Location_Type     : Factor w/ 3 levels "Tier 1","Tier 2",..: 1 3 1 3 3 3 3 3 2 2 ...
##  $ Outlet_Type              : Factor w/ 4 levels "Grocery Store",..: 2 3 2 1 2 3 2 4 2 2 ...
##  $ Item_Outlet_Sales        : num  3735 443 2097 732 995 ...
```

```r
# Remove item_identifier
data <- data[, !(colnames(data) %in% c("Item_Identifier"))]

# Convert labels to factor and back to variables
data <- data.matrix(data)

data <- data.frame(data)

# Omit rows with NA
data <- na.omit(data)
str(data)
```

```
## 'data.frame':    7060 obs. of  11 variables:
##  $ Item_Weight              : num  9.3 5.92 17.5 19.2 8.93 ...
##  $ Item_Fat_Content         : num  3 5 3 5 3 5 5 5 5 3 ...
##  $ Item_Visibility          : num  0.016 0.0193 0.0168 0 0 ...
##  $ Item_Type                : num  5 15 11 7 10 1 14 6 6 7 ...
##  $ Item_MRP                 : num  249.8 48.3 141.6 182.1 53.9 ...
##  $ Outlet_Identifier        : num  10 4 10 1 2 4 2 8 3 10 ...
##  $ Outlet_Establishment_Year: num  1999 2009 1999 1998 1987 ...
##  $ Outlet_Size              : num  3 3 3 1 2 3 2 1 1 3 ...
##  $ Outlet_Location_Type     : num  1 3 1 3 3 3 3 2 2 1 ...
##  $ Outlet_Type              : num  2 3 2 1 2 3 2 2 2 2 ...
##  $ Item_Outlet_Sales        : num  3735 443 2097 732 995 ...
##  - attr(*, "na.action")= 'omit' Named int  8 19 22 24 30 37 39 40 50 60 ...
##   ..- attr(*, "names")= chr  "8" "19" "22" "24" ...
```

Part b

```r
set.seed(1)

sample <- sample.int(n = nrow(data), size = floor(.5*nrow(data)), replace = F)
train <- data[sample, ]
test  <- data[-sample, ]

train.x <- data.matrix(train[, !(colnames(train) %in% c("Item_Outlet_Sales"))])
train.y <- data.matrix(train["Item_Outlet_Sales"])
```

```
test.x <- data.matrix(test[, !(colnames(test) %in% c("Item_Outlet_Sales"))])
test.y <- data.matrix(test["Item_Outlet_Sales"])

ridgefit <- cv.glmnet(train.x, train.y, alpha=0, nfold=10)
lassofit <- cv.glmnet(train.x, train.y, alpha=1, nfold=10)
ridgefit$lambda.min
```

```
## [1] 95.99262
```

```
ridgefit$lambda.1se
```

```
## [1] 425.307
```

```
lassofit$lambda.min
```

```
## [1] 1.183442
```

```
lassofit$lambda.1se
```

```
## [1] 85.45408
```

```
ridgemin <- glmnet(train.x, train.y, alpha=0, lambda = ridgefit$lambda.min)
ridge1se <- glmnet(train.x, train.y, alpha=0, lambda = ridgefit$lambda.1se)
lassomin <- glmnet(train.x, train.y, alpha=1, lambda = lassofit$lambda.min)
lasso1se <- glmnet(train.x, train.y, alpha=1, lambda = lassofit$lambda.1se)

ridgemin$beta
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                                      s0
## Item_Weight                   0.1762635
## Item_Fat_Content             19.5768921
## Item_Visibility           -1231.2263667
## Item_Type                     3.6062646
## Item_MRP                     14.3317388
## Outlet_Identifier             6.9592473
## Outlet_Establishment_Year   -21.9581608
## Outlet_Size                 -69.9860179
## Outlet_Location_Type       -372.2767336
## Outlet_Type                 758.3360663
```

```
ridge1se$beta
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                                      s0
## Item_Weight                   0.3925739
## Item_Fat_Content             17.1218263
## Item_Visibility           -1314.5956897
## Item_Type                     3.8361709
```

```
## Item_MRP                     11.9517855
## Outlet_Identifier            26.0041785
## Outlet_Establishment_Year   -12.7859554
## Outlet_Size                 -20.7443419
## Outlet_Location_Type       -218.2387583
## Outlet_Type                 512.0470109
```

```
lassomin$beta
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                                    s0
## Item_Weight                         .
## Item_Fat_Content             18.459888
## Item_Visibility           -1088.094124
## Item_Type                     3.134516
## Item_MRP                     15.198564
## Outlet_Identifier           -18.389526
## Outlet_Establishment_Year   -27.116633
## Outlet_Size                 -94.851984
## Outlet_Location_Type       -511.006957
## Outlet_Type                 910.338846
```

```
lasso1se$beta
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                                    s0
## Item_Weight                         .
## Item_Fat_Content                    .
## Item_Visibility           -123.42157
## Item_Type                           .
## Item_MRP                    13.91709
## Outlet_Identifier                   .
## Outlet_Establishment_Year           .
## Outlet_Size                         .
## Outlet_Location_Type      -233.41122
## Outlet_Type                399.36890
```

```
y_ridgemin <- predict.glmnet(ridgemin, test.x)
y_ridge1se <- predict.glmnet(ridge1se, test.x)
y_lassomin <- predict.glmnet(lassomin, test.x)
y_lasso1se <- predict.glmnet(lasso1se, test.x)

sqrt(sum((test.y-y_ridgemin)^2))
```

```
## [1] 65644.6
```

```
sqrt(sum((test.y-y_ridge1se)^2))
```

```
## [1] 67173.87
```

```r
sqrt(sum((test.y-y_lassomin)^2))
```

```
## [1] 65407.49
```

```r
sqrt(sum((test.y-y_lasso1se)^2))
```

```
## [1] 66894.69
```

Comparing mse of all predictions, the best model is lasso regresion with minimal lambda 1.183442