Take Home Exams

Logan Liu (gl22453)

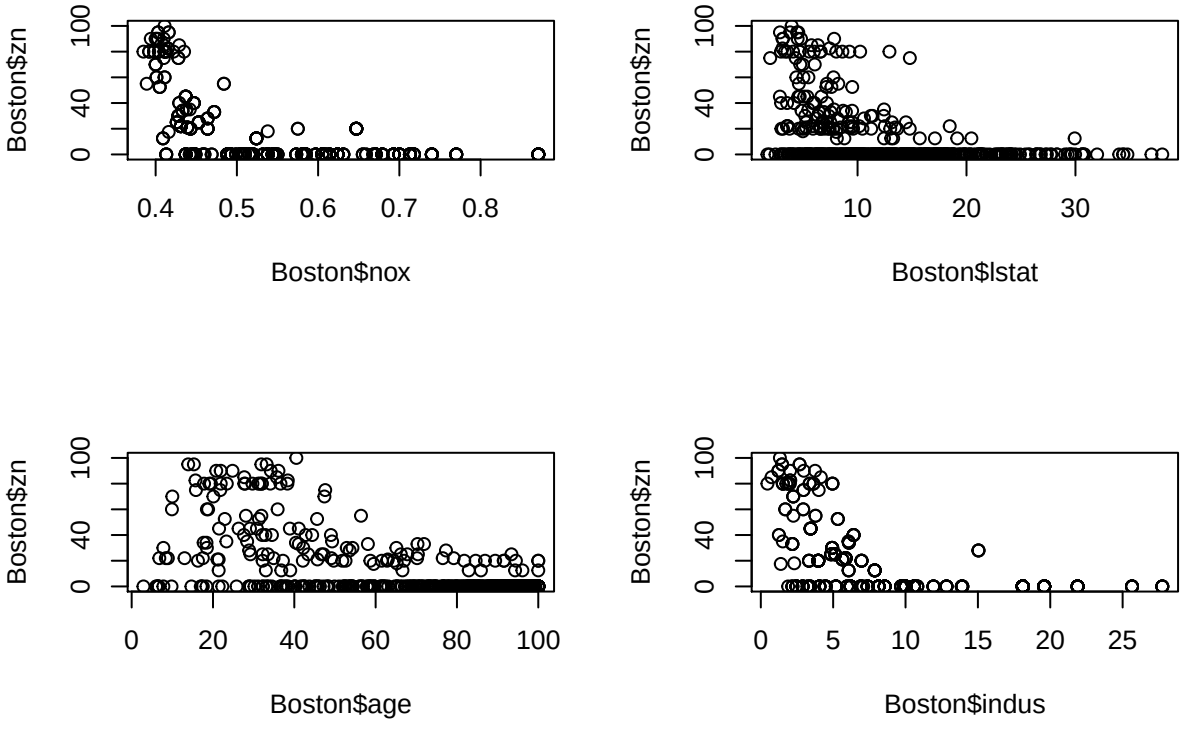
8/9/2019

Book Problems

Chapter 2: #10

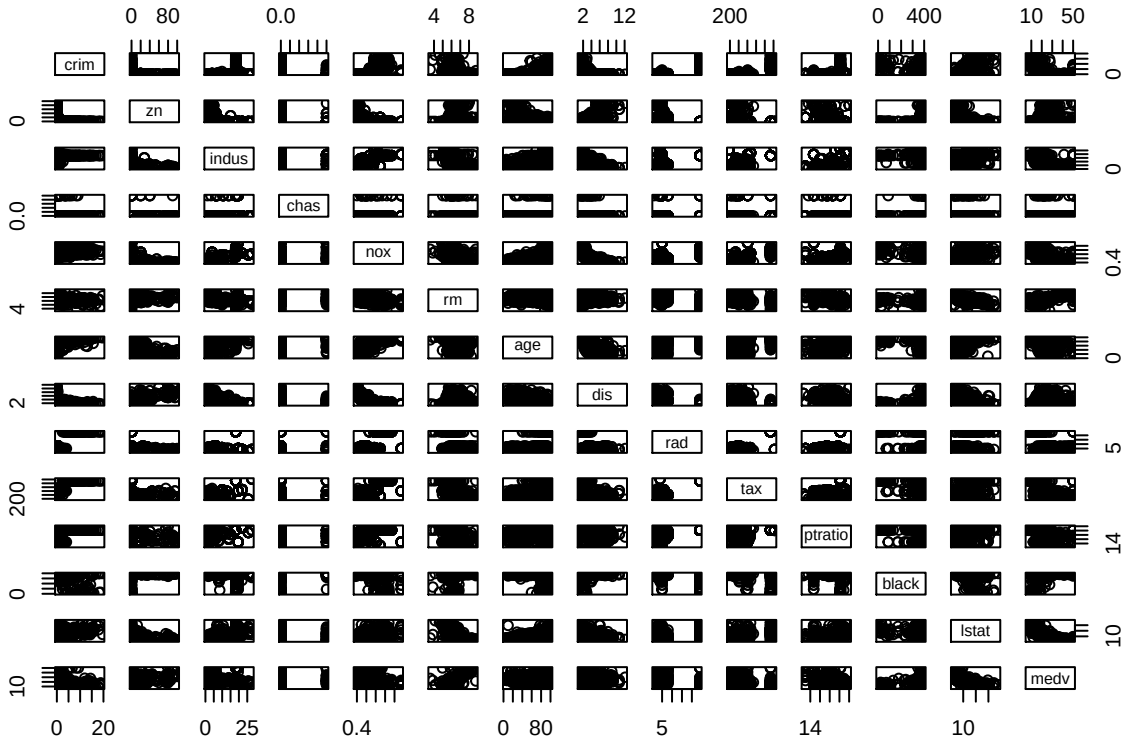
(a) dim(Boston)
The Boston data frame has 506 rows and 14 columns.
This data frame contains the following columns: crim per capita crime rate by town. zn proportion of residential land zoned for lots over 25,000 sq.ft. indus proportion of non-retail business acres per town. chas Charles River dummy variable (= 1 if tract bounds river; 0 otherwise). nox nitrogen oxides concentration (parts per 10 million). rm average number of rooms per dwelling. age proportion of owner-occupied units built prior to 1940. dis weighted mean of distances to five Boston employment centres. rad index of accessibility to radial highways. tax full-value property-tax rate per $10,000. ptratio pupil-teacher ratio by town. black 1000(Bk - 0.63)ˆ2 where Bk is the proportion of blacks by town. lstat lower status of the population (percent). medv median value of owner-occupied homes in $1000s.

(b) Nitrogen oxides concentration, lower status of the population, proportion of owner-occupied units built prior to 1940, and proportion of non-retail business acres per town are all predictors of proportion of residential land zoned for lots over 25,000 sq.ft.

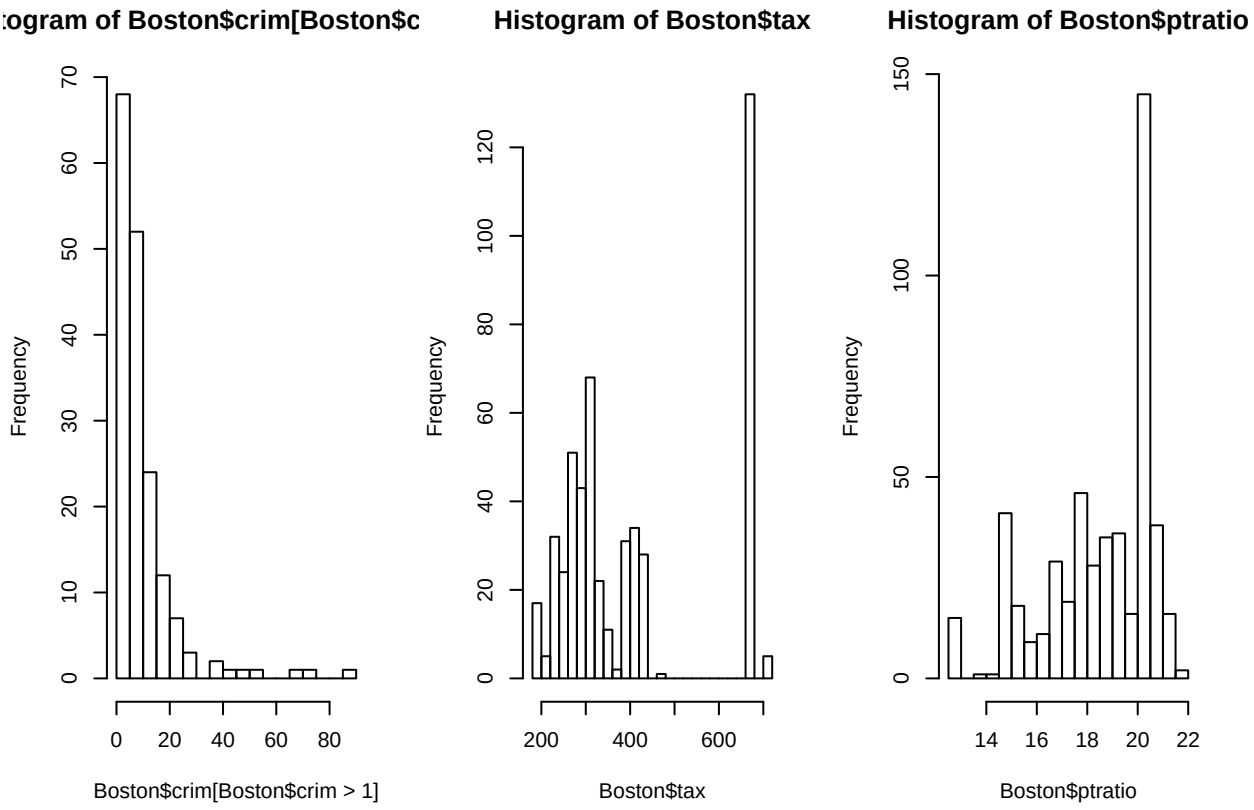


(c) There is a relationship between crim and nox, rm, age, dis, lstat and medv.

Crime rate is high when high nitrogen oxides concentration, low average number of rooms per dwelling, high tax rate, short distances to five Boston employment centres, little proportion of owner-occupied units built prior to 1940.



(d) There are 18 suburbs with high crime rates more than 20.
When tax is 666, there is a very high crime rate.
The higher pupil-teacher ratios, the higher crime rate. But not very correlated.



(e) 35
(f) 19.05
(g) t(subset(Boston, medv == min(Boston$medv)))
The suburb with the lowest median value is 398. Relative to the other towns, this suburb has high crim, zn below quantile 75%, above mean indus, does not bound the Charles river, above mean nox, rm below quantile 25%, maximum age, dis near to the minimum value, maximum rad, tax and ptratio in quantile 75%, black maximum and lstat above quantile 75%.
(h) 64
13
crim is lower, indus proportion is lower, lstat is lower.

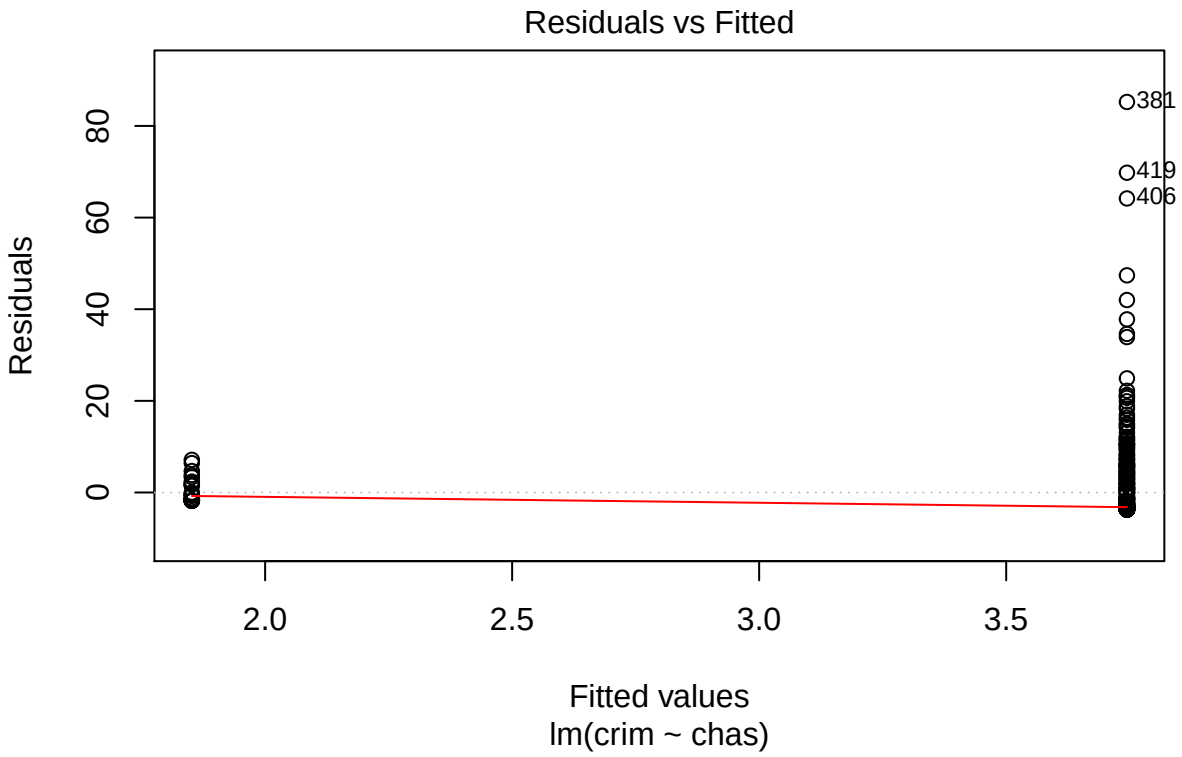
```
##       crim                zn              indus              chas        
##  Min.   :0.02009   Min.   : 0.00   Min.   : 2.680   Min.   :0.0000  
##  1st Qu.:0.33147   1st Qu.: 0.00   1st Qu.: 3.970   1st Qu.:0.0000  
##  Median :0.52014   Median : 0.00   Median : 6.200   Median :0.0000  
##  Mean   :0.71879   Mean   :13.62   Mean   : 7.078   Mean   :0.1538  
##  3rd Qu.:0.57834   3rd Qu.:20.00   3rd Qu.: 6.200   3rd Qu.:0.0000  
##  Max.   :3.47428   Max.   :95.00   Max.   :19.580   Max.   :1.0000  
##       nox               rm              age              dis        
##  Min.   :0.4161   Min.   :8.034   Min.   : 8.40   Min.   :1.801  
##  1st Qu.:0.5040   1st Qu.:8.247   1st Qu.:70.40   1st Qu.:2.288  
##  Median :0.5070   Median :8.297   Median :78.30   Median :2.894  
##  Mean   :0.5392   Mean   :8.349   Mean   :71.54   Mean   :3.430  
##  3rd Qu.:0.6050   3rd Qu.:8.398   3rd Qu.:86.50   3rd Qu.:3.652  
##  Max.   :0.7180   Max.   :8.780   Max.   :93.90   Max.   :8.907  
##       rad              tax             ptratio          black       
##  Min.   : 2.000   Min.   :224.0   Min.   :13.00   Min.   :354.6
```

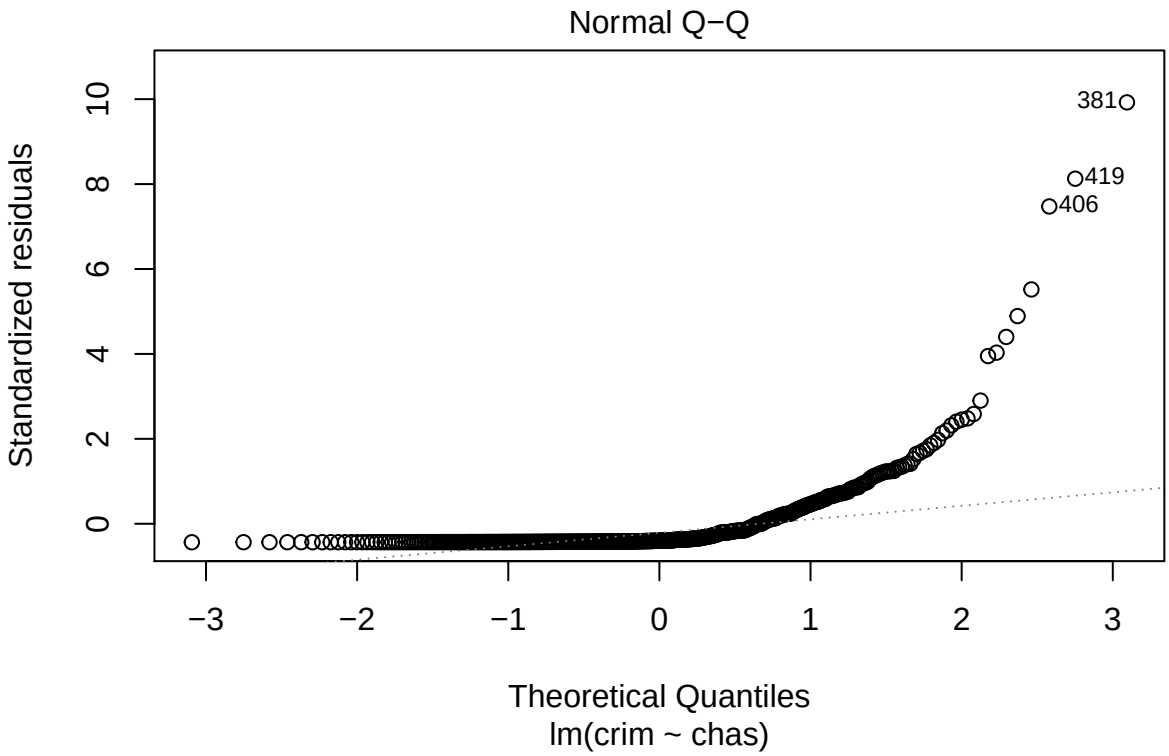
```
##  1st Qu.: 5.000   1st Qu.:264.0   1st Qu.:14.70   1st Qu.:384.5
##  Median : 7.000   Median :307.0   Median :17.40   Median :386.9
##  Mean   : 7.462   Mean   :325.1   Mean   :16.36   Mean   :385.2
##  3rd Qu.: 8.000   3rd Qu.:307.0   3rd Qu.:17.40   3rd Qu.:389.7
##  Max.   :24.000   Max.   :666.0   Max.   :20.20   Max.   :396.9
##      lstat           medv
##  Min.   :2.47   Min.   :21.9
##  1st Qu.:3.32   1st Qu.:41.7
##  Median :4.14   Median :48.3
##  Mean   :4.31   Mean   :44.2
##  3rd Qu.:5.12   3rd Qu.:50.0
##  Max.   :7.44   Max.   :50.0
```

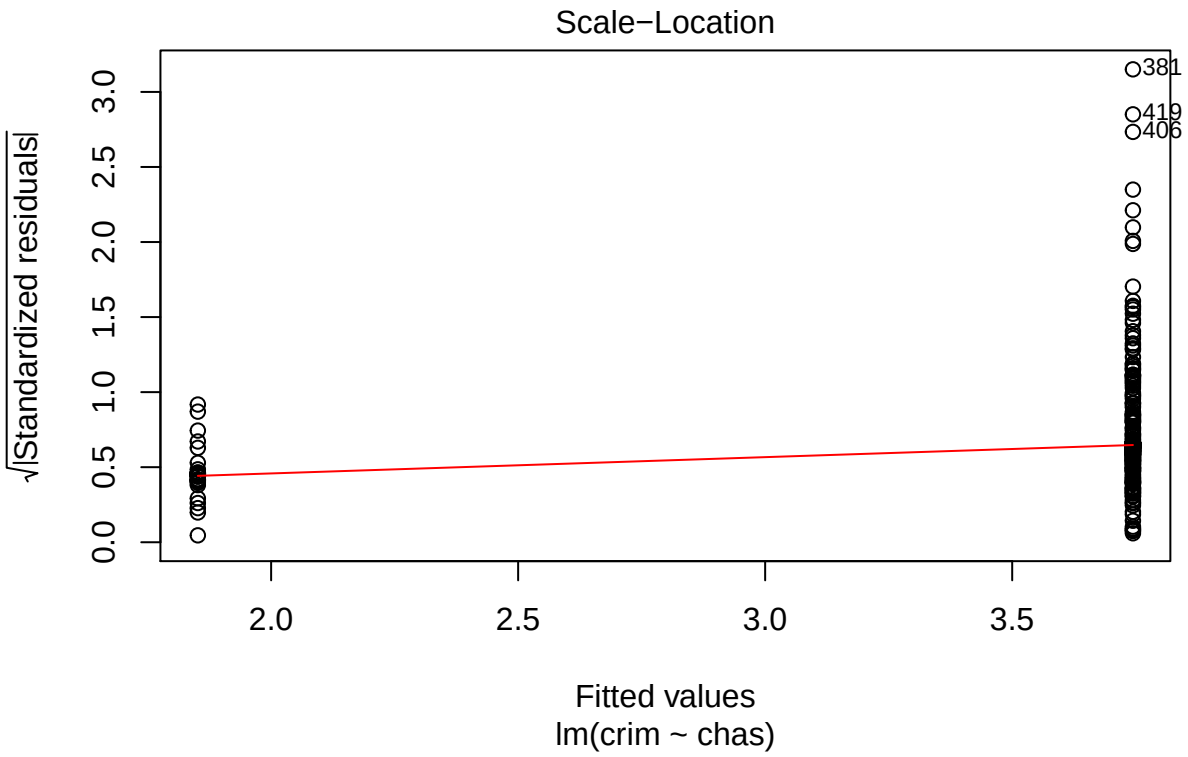
Chapter 3: #15

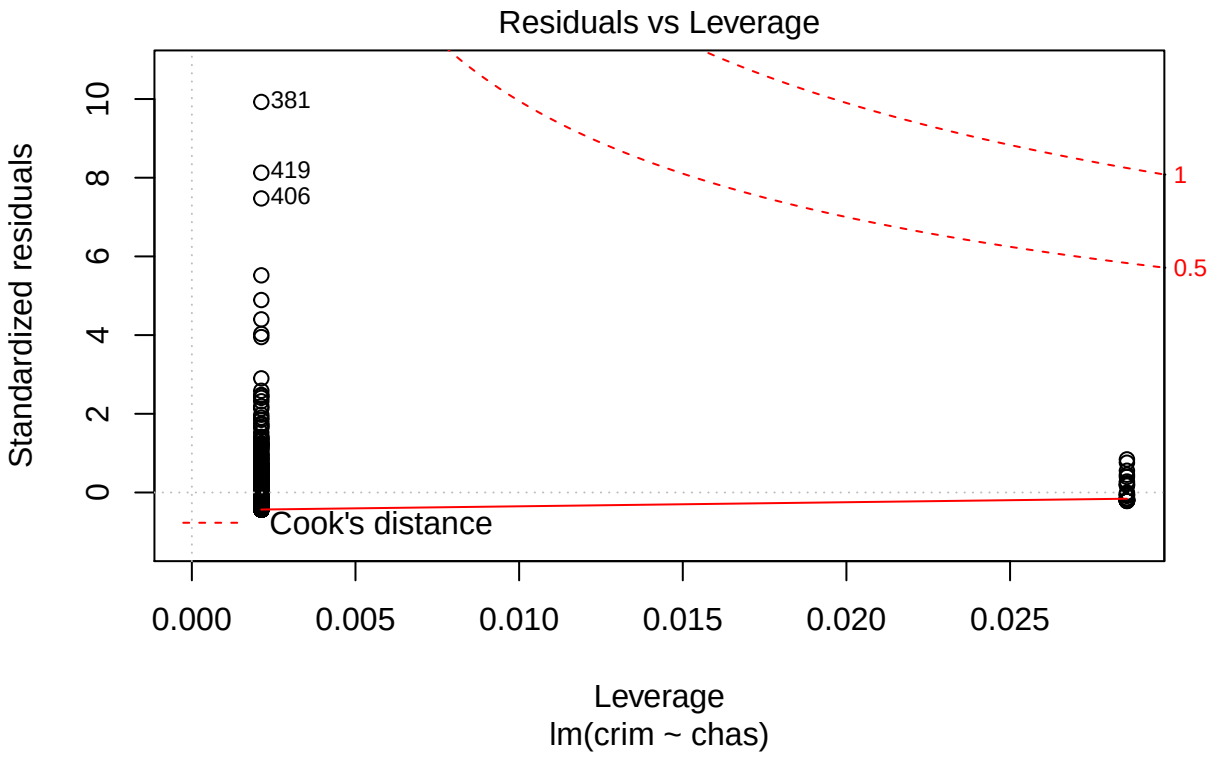
(a) Every predictor except chas has a statistically significant association with crim.

```
##
## Call:
## lm(formula = crim ~ chas)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.738 -3.661 -3.435  0.018 85.232
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.7444     0.3961   9.453   <2e-16 ***
## chas         -1.8928     1.5061  -1.257    0.209
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.597 on 504 degrees of freedom
## Multiple R-squared:  0.003124,   Adjusted R-squared:  0.001146
## F-statistic: 1.579 on 1 and 504 DF,  p-value: 0.2094
```


Residuals vs Fitted
Residuals
0
20
40
60
80
381
419
406
2.0
2.5
3.0
3.5
Fitted values
lm(crim ~ chas)


Normal Q−Q
Standardized residuals
0
2
4
6
8
10
381
419
406
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(crim ~ chas)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
381
419
406
2.0
2.5
3.0
3.5
Fitted values
lm(crim ~ chas)

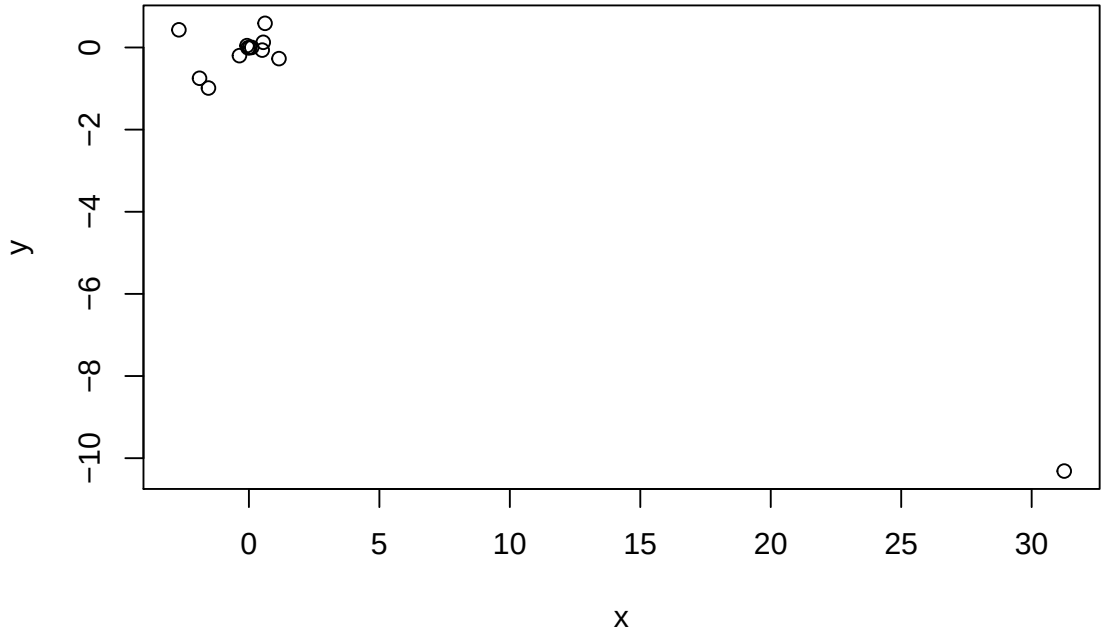


(b) We can reject the null-hypothesis for "zn", "dis", "rad", "black" and "medv". Because their p values are less than 0.05.

```
##
## Call:
## lm(formula = crim ~ ., data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.924 -2.120 -0.353  1.019 75.051
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.033228   7.234903   2.354 0.018949 *
## zn            0.044855   0.018734   2.394 0.017025 *
## indus        -0.063855   0.083407  -0.766 0.444294
## chas         -0.749134   1.180147  -0.635 0.525867
## nox         -10.313535   5.275536  -1.955 0.051152 .
## rm            0.430131   0.612830   0.702 0.483089
## age           0.001452   0.017925   0.081 0.935488
## dis          -0.987176   0.281817  -3.503 0.000502 ***
## rad           0.588209   0.088049   6.680 6.46e-11 ***
## tax          -0.003780   0.005156  -0.733 0.463793
## ptratio      -0.271081   0.186450  -1.454 0.146611
## black        -0.007538   0.003673  -2.052 0.040702 *
```

```
## lstat         0.126211   0.075725   1.667 0.096208 .
## medv         -0.198887   0.060516  -3.287 0.001087 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.439 on 492 degrees of freedom
## Multiple R-squared:  0.454,  Adjusted R-squared:  0.4396
## F-statistic: 31.47 on 13 and 492 DF,  p-value: < 2.2e-16
```

(c) It differs significantly between the individual and multiple regression. Since in the former the coefficient is the average change in the response from a unit change in the predictor completely ignoring the other predictors. In the latter case, the coefficient is the average change in the response from a unit change in the predictor while holding the other predictor fixed.



(d) We can find evidence of a non-linear association, cubic type, between INDUS, NOX, AGE, DIS, PTRATIO and MEDV.

```
##
## Call:
## lm(formula = crim ~ poly(zn, 3))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.821 -4.614 -1.294  0.473 84.130
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.6135     0.3722   9.709  < 2e-16 ***
## poly(zn, 3)1 -38.7498     8.3722  -4.628  4.7e-06 ***
```

```
## poly(zn, 3)2  23.9398     8.3722   2.859  0.00442 **
## poly(zn, 3)3 -10.0719     8.3722  -1.203  0.22954
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.372 on 502 degrees of freedom
## Multiple R-squared:  0.05824,    Adjusted R-squared:  0.05261
## F-statistic: 10.35 on 3 and 502 DF,  p-value: 1.281e-06


##
## Call:
## lm(formula = crim ~ poly(indus, 3))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.278 -2.514  0.054  0.764 79.713
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)        3.614      0.330  10.950  < 2e-16 ***
## poly(indus, 3)1   78.591      7.423  10.587  < 2e-16 ***
## poly(indus, 3)2  -24.395      7.423  -3.286  0.00109 **
## poly(indus, 3)3  -54.130      7.423  -7.292  1.2e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.423 on 502 degrees of freedom
## Multiple R-squared:  0.2597, Adjusted R-squared:  0.2552
## F-statistic: 58.69 on 3 and 502 DF,  p-value: < 2.2e-16


##
## Call:
## lm(formula = crim ~ poly(nox, 3))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.110 -2.068 -0.255  0.739 78.302
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.3216  11.237  < 2e-16 ***
## poly(nox, 3)1  81.3720     7.2336  11.249  < 2e-16 ***
## poly(nox, 3)2 -28.8286     7.2336  -3.985 7.74e-05 ***
## poly(nox, 3)3 -60.3619     7.2336  -8.345 6.96e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.234 on 502 degrees of freedom
## Multiple R-squared:  0.297,  Adjusted R-squared:  0.2928
## F-statistic: 70.69 on 3 and 502 DF,  p-value: < 2.2e-16


##
## Call:
```

```
## lm(formula = crim ~ poly(rm, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.485  -3.468  -2.221  -0.015  87.219
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.6135     0.3703   9.758  < 2e-16 ***
## poly(rm, 3)1 -42.3794     8.3297  -5.088 5.13e-07 ***
## poly(rm, 3)2  26.5768     8.3297   3.191  0.00151 **
## poly(rm, 3)3  -5.5103     8.3297  -0.662  0.50858
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.33 on 502 degrees of freedom
## Multiple R-squared:  0.06779,    Adjusted R-squared:  0.06222
## F-statistic: 12.17 on 3 and 502 DF,  p-value: 1.067e-07

##
## Call:
## lm(formula = crim ~ poly(age, 3))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.762 -2.673 -0.516  0.019 82.842
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.3485  10.368  < 2e-16 ***
## poly(age, 3)1  68.1820     7.8397   8.697  < 2e-16 ***
## poly(age, 3)2  37.4845     7.8397   4.781 2.29e-06 ***
## poly(age, 3)3  21.3532     7.8397   2.724  0.00668 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.84 on 502 degrees of freedom
## Multiple R-squared:  0.1742, Adjusted R-squared:  0.1693
## F-statistic: 35.31 on 3 and 502 DF,  p-value: < 2.2e-16

##
## Call:
## lm(formula = crim ~ poly(dis, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.757  -2.588   0.031   1.267  76.378
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.3259  11.087  < 2e-16 ***
## poly(dis, 3)1 -73.3886     7.3315 -10.010  < 2e-16 ***
## poly(dis, 3)2  56.3730     7.3315   7.689 7.87e-14 ***
## poly(dis, 3)3 -42.6219     7.3315  -5.814 1.09e-08 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.331 on 502 degrees of freedom
## Multiple R-squared:  0.2778, Adjusted R-squared:  0.2735
## F-statistic: 64.37 on 3 and 502 DF,  p-value: < 2.2e-16

##
## Call:
## lm(formula = crim ~ poly(rad, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.381  -0.412  -0.269   0.179  76.217
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.2971  12.164  < 2e-16 ***
## poly(rad, 3)1 120.9074     6.6824  18.093  < 2e-16 ***
## poly(rad, 3)2  17.4923     6.6824   2.618  0.00912 **
## poly(rad, 3)3   4.6985     6.6824   0.703  0.48231
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.682 on 502 degrees of freedom
## Multiple R-squared:    0.4,  Adjusted R-squared:  0.3965
## F-statistic: 111.6 on 3 and 502 DF,  p-value: < 2.2e-16

##
## Call:
## lm(formula = crim ~ poly(tax, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.273  -1.389   0.046   0.536  76.950
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.3047  11.860  < 2e-16 ***
## poly(tax, 3)1 112.6458     6.8537  16.436  < 2e-16 ***
## poly(tax, 3)2  32.0873     6.8537   4.682 3.67e-06 ***
## poly(tax, 3)3  -7.9968     6.8537  -1.167    0.244
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.854 on 502 degrees of freedom
## Multiple R-squared:  0.3689, Adjusted R-squared:  0.3651
## F-statistic:  97.8 on 3 and 502 DF,  p-value: < 2.2e-16

##
## Call:
## lm(formula = crim ~ poly(ptratio, 3))
##
```

```
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.833 -4.146 -1.655  1.408 82.697
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)          3.614      0.361  10.008  < 2e-16 ***
## poly(ptratio, 3)1   56.045      8.122   6.901 1.57e-11 ***
## poly(ptratio, 3)2   24.775      8.122   3.050  0.00241 **
## poly(ptratio, 3)3  -22.280      8.122  -2.743  0.00630 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.122 on 502 degrees of freedom
## Multiple R-squared:  0.1138, Adjusted R-squared:  0.1085
## F-statistic: 21.48 on 3 and 502 DF,  p-value: 4.171e-13
```

```
##
## Call:
## lm(formula = crim ~ poly(black, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.096  -2.343  -2.128  -1.439  86.790
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.6135     0.3536  10.218   <2e-16 ***
## poly(black, 3)1 -74.4312     7.9546  -9.357   <2e-16 ***
## poly(black, 3)2   5.9264     7.9546   0.745    0.457
## poly(black, 3)3  -4.8346     7.9546  -0.608    0.544
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.955 on 502 degrees of freedom
## Multiple R-squared:  0.1498, Adjusted R-squared:  0.1448
## F-statistic: 29.49 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
##
## Call:
## lm(formula = crim ~ poly(lstat, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.234  -2.151  -0.486   0.066  83.353
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.6135     0.3392  10.654   <2e-16 ***
## poly(lstat, 3)1  88.0697     7.6294  11.543   <2e-16 ***
## poly(lstat, 3)2  15.8882     7.6294   2.082   0.0378 *
## poly(lstat, 3)3 -11.5740     7.6294  -1.517   0.1299
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 7.629 on 502 degrees of freedom
## Multiple R-squared:  0.2179, Adjusted R-squared:  0.2133
## F-statistic: 46.63 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
##
## Call:
## lm(formula = crim ~ poly(medv, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.427  -1.976  -0.437   0.439  73.655
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.614      0.292  12.374  < 2e-16 ***
## poly(medv, 3)1  -75.058      6.569 -11.426  < 2e-16 ***
## poly(medv, 3)2   88.086      6.569  13.409  < 2e-16 ***
## poly(medv, 3)3  -48.033      6.569  -7.312 1.05e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.569 on 502 degrees of freedom
## Multiple R-squared:  0.4202, Adjusted R-squared:  0.4167
## F-statistic: 121.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

Chapter 6: #9

(a) set.seed(1)
Functions: sample

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

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

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

(b) Test error is 1246301
Functions: lm

```
## [1] 1246301
```

(c) lambda: 18.74; MSE: 1608859
Functions: model.matrix, cv.glmnet

```
## [1] 24.77076
```

```
## [1] 1305614
```

(d) lambda: 21.54; MSE: 1135660

```
## [1] 16.29751
```

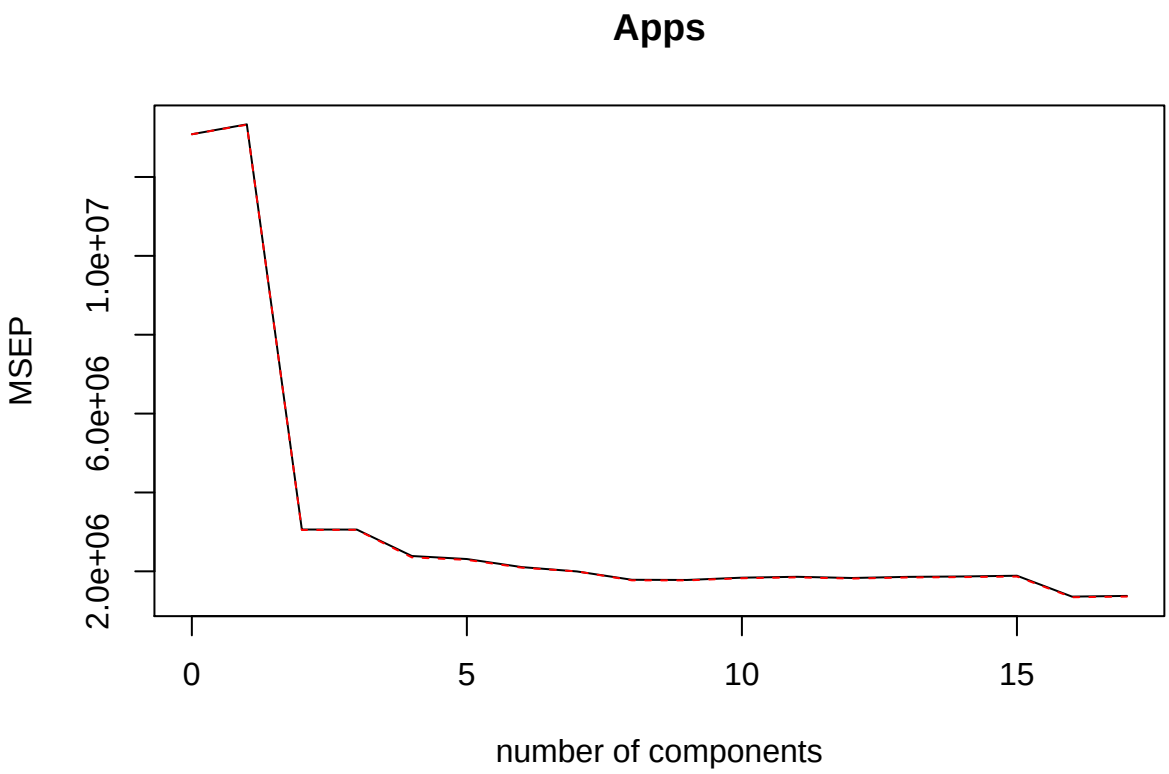
```
## [1] 1228865
```

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                          1
## (Intercept) -5.676511e+02
## (Intercept)  .
## PrivateYes  -4.484610e+02
## Accept       1.486731e+00
## Enroll      -3.017176e-01
## Top10perc    3.755013e+01
## Top25perc   -5.090937e+00
## F.Undergrad  .
## P.Undergrad  3.020278e-02
## Outstate    -6.518772e-02
## Room.Board   1.318802e-01
## Books        .
## Personal     7.452199e-03
## PhD         -6.397825e+00
## Terminal    -3.202102e+00
## S.F.Ratio    7.410779e+00
## perc.alumni -8.009749e-01
## Expend       7.168949e-02
## Grad.Rate    5.957452e+00
```

(e) MSE: 1723100
Functions: pcr

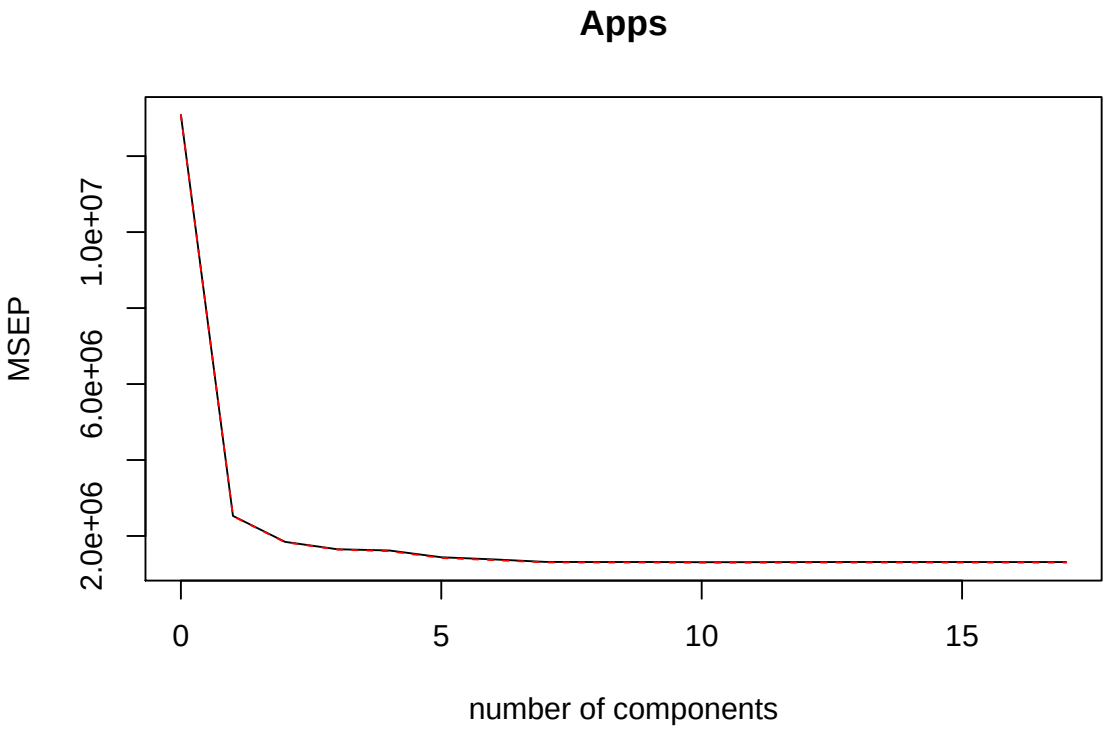
```
##
## Attaching package: 'pls'
```

```
## The following object is masked from 'package:stats':
##
##     loadings
```




```
## [1] 2701381
```

(f) MSE: 1131661
Functions: plsr




```
## [1] 1251891
```

(g) Based on the R square,

Chapter 6: #11

(a) Best subset selection: MSE - 50.43627
The best model is the one contains 9 variables.
Functions: regsubsets()

```
##  [1] 53.58865 54.37254 52.66064 52.50837 51.53113 51.12540 51.04084
##  [8] 50.69984 50.43627 50.52382 50.71664 50.68209 50.65678
```

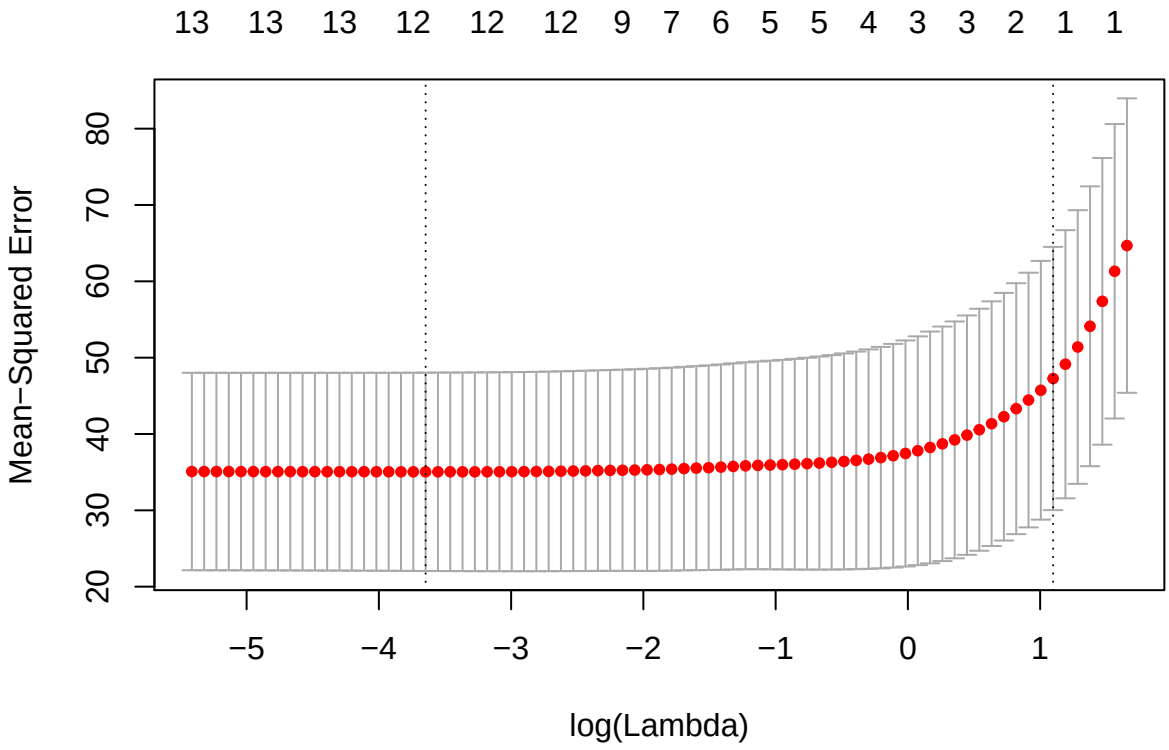
```
## [1] 9
```

```
## [1] 50.43627
```

Lasso: lambda - 0.02608302; MSE - 50.75568
Functions: cv.glmnet()

```
## [1] 0.02608302
```

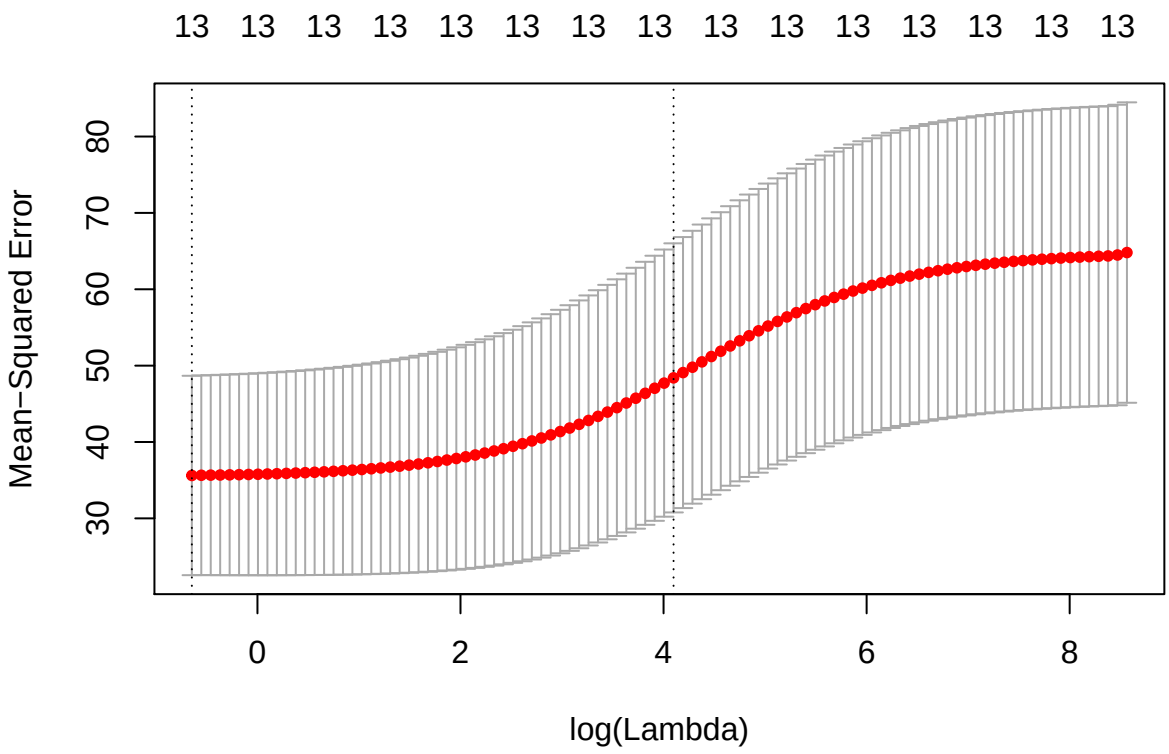
```
## [1] 50.75568
```



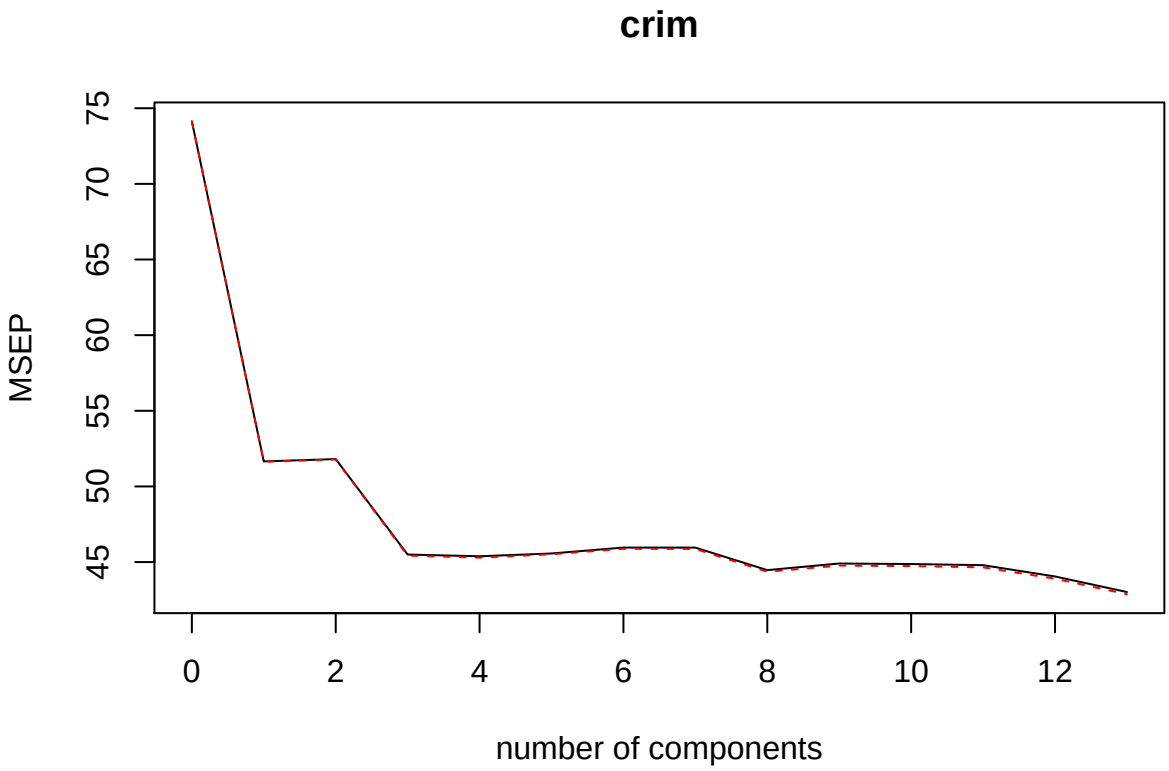
Ridge: lambda - 0.5240686; MSE - 51.46284
Functions: cv.glmnet()

```
## [1] 0.5240686
```

```
## [1] 51.46284
```



PCR: MSE - 51.68033
Functions: pcr()




```
## [1] 51.68033
```

(b) Best subset selection method has the lowest cross-validation error, which is 50.43627.

(c) No, the model chosen by the best subset selection method has only 9 predictors.

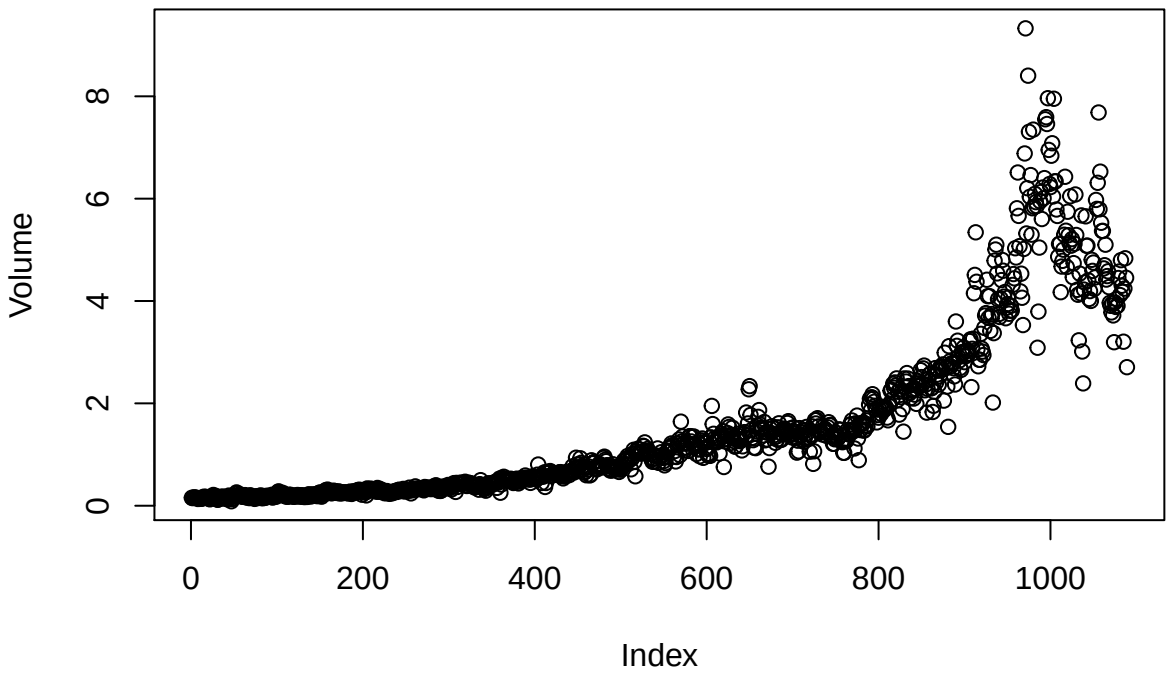
Chapter 4: #10

(a) As one would expect, the correlations between the lag variables and today's returns are close to zero. In other words, there appears to be little correlation between today's returns and previous days' returns. The only substantial correlation is between Year and Volume. By plotting the data we see that Volume is increasing over time.
Functions: summary(), cor ()

```
##       Year           Lag1                Lag2                Lag3          
##  Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950  
##  1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580  
##  Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410  
##  Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472  
##  3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090  
##  Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260  
##       Lag4                Lag5              Volume       
##  Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747  
##  1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202  
##  Median :  0.2380   Median :  0.2340   Median :1.00268
```

```
##  Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462
##  3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373
##  Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821
##      Today          Direction
##  Min.   :-18.1950   Down:484
##  1st Qu.: -1.1540   Up  :605
##  Median :  0.2410
##  Mean   :  0.1499
##  3rd Qu.:  1.4050
##  Max.   : 12.0260
```

```
##               Year         Lag1        Lag2        Lag3         Lag4
## Year    1.00000000 -0.032289274 -0.03339001 -0.03000649 -0.031127923
## Lag1   -0.03228927  1.000000000 -0.07485305  0.05863568 -0.071273876
## Lag2   -0.03339001 -0.074853051  1.00000000 -0.07572091  0.058381535
## Lag3   -0.03000649  0.058635682 -0.07572091  1.00000000 -0.075395865
## Lag4   -0.03112792 -0.071273876  0.05838153 -0.07539587  1.000000000
## Lag5   -0.03051910 -0.008183096 -0.07249948  0.06065717 -0.075675027
## Volume  0.84194162 -0.064951313 -0.08551314 -0.06928771 -0.061074617
## Today  -0.03245989 -0.075031842  0.05916672 -0.07124364 -0.007825873
##                Lag5      Volume        Today
## Year   -0.030519101  0.84194162 -0.032459894
## Lag1   -0.008183096 -0.06495131 -0.075031842
## Lag2   -0.072499482 -0.08551314  0.059166717
## Lag3    0.060657175 -0.06928771 -0.071243639
## Lag4   -0.075675027 -0.06107462 -0.007825873
## Lag5    1.000000000 -0.05851741  0.011012698
## Volume -0.058517414  1.00000000 -0.033077783
## Today   0.011012698 -0.03307778  1.000000000
```



(b) Functions: The smallest p-value here is associated with Lag2. The positive coefficient for this predictor suggests that if the market had a positive return yesterday, then it is more likely to go up today.
Functions: glm()

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial, data = Weekly)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6949  -1.2565   0.9913   1.0849   1.4579
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26686    0.08593   3.106   0.0019 **
## Lag1        -0.04127    0.02641  -1.563   0.1181
## Lag2         0.05844    0.02686   2.175   0.0296 *
## Lag3        -0.01606    0.02666  -0.602   0.5469
## Lag4        -0.02779    0.02646  -1.050   0.2937
## Lag5        -0.01447    0.02638  -0.549   0.5833
## Volume      -0.02274    0.03690  -0.616   0.5377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
## AIC: 1500.4
##
## Number of Fisher Scoring iterations: 4
```

(c) The diagonal elements of the confusion matrix indicate correct predictions, while the off-diagonals represent incorrect predictions. Hence our model correctly predicted that the market would go up on 557 days and that it would go down on 54 days, for a total of 557 + 54 = 611 correct predictions. Correct rate: 56.10652% predictions.
Functions: table()

```
##         Direction
## glm.pred Down  Up
##     Down   54  48
##     Up    430 557
```

```
## [1] 0.5610652
```

(d) Our model correctly predicted that the market would go up on 56 days and that it would go down on 9 days, for a total of 56 + 9 = 65 correct predictions. Correct rate: 62.5%
Functions: glm()

```
##
## Call:
## glm(formula = Direction ~ Lag2, family = binomial, data = Weekly,
##     subset = train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.536  -1.264   1.021   1.091   1.368
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.20326    0.06428   3.162  0.00157 **
## Lag2         0.05810    0.02870   2.024  0.04298 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1354.7  on 984  degrees of freedom
## Residual deviance: 1350.5  on 983  degrees of freedom
## AIC: 1354.5
##
## Number of Fisher Scoring iterations: 4
```

```
##          Direction.20092010
## pred.glm2 Down Up
##      Down    9  5
##      Up     34 56
```

```
## [1] 0.625
```

(g) Our model correctly predicted that the market would go up on 31 days and that it would go down on 21 days, for a total of 21 + 31 = 52 correct predictions. Correct rate: 50%
Functions: knn() k=1

```
##         Direction.20092010
## pred.knn Down Up
##     Down   21 30
##     Up     22 31
```

```
## [1] 0.5
```

(h) Logistic regression has better test correction rate than KNN.

(i) The results have improved slightly. But increasing K further turns out to provide no further improvements. It appears that for this data, logistic regression provides the best results of the methods that we have examined so far.

KNN k =10

```
##          Direction.20092010
## pred.knn2 Down Up
##      Down   17 18
##      Up     26 43
```

```
## [1] 0.5769231
```

KNN k =100

```
##          Direction.20092010
## pred.knn3 Down Up
##      Down    9 12
##      Up     34 49
```

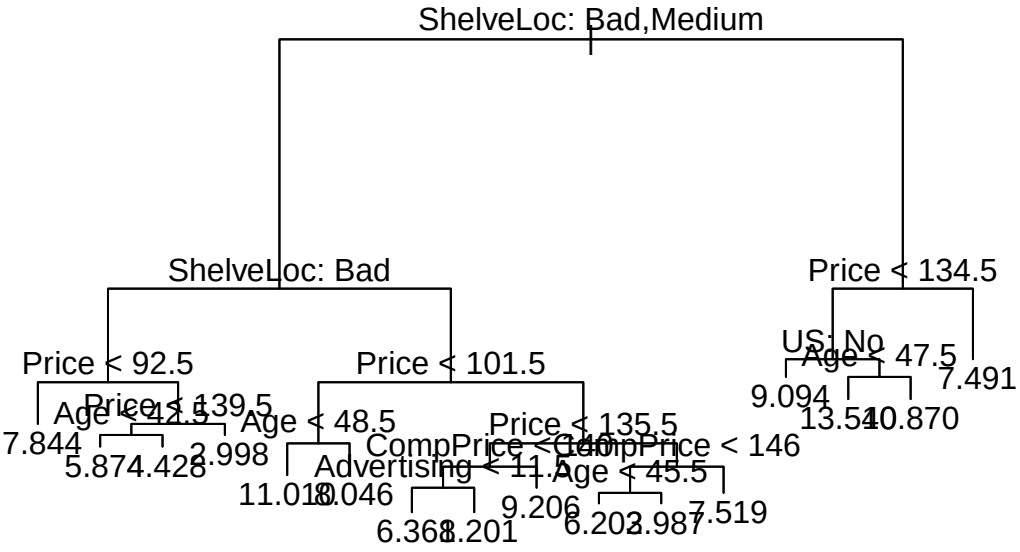
```
## [1] 0.5576923
```

Chapter 8: #8

(a) train = sample(1:nrow(Carseats), nrow(Carseats) / 2)

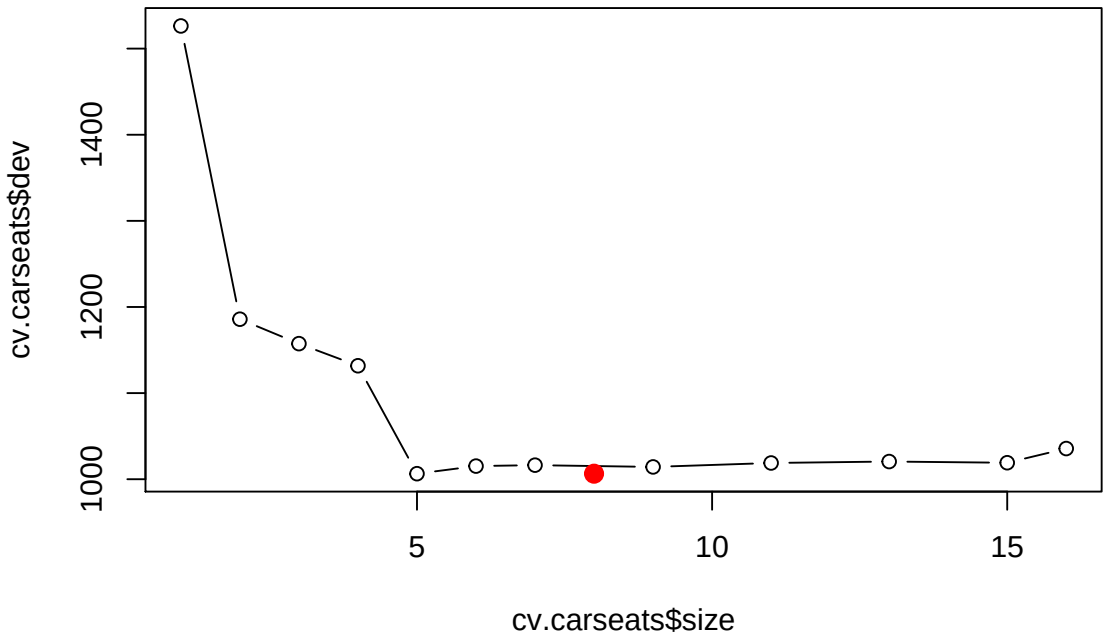
(b) MSE: 4.784151
Functions: tree(), plot(), text()

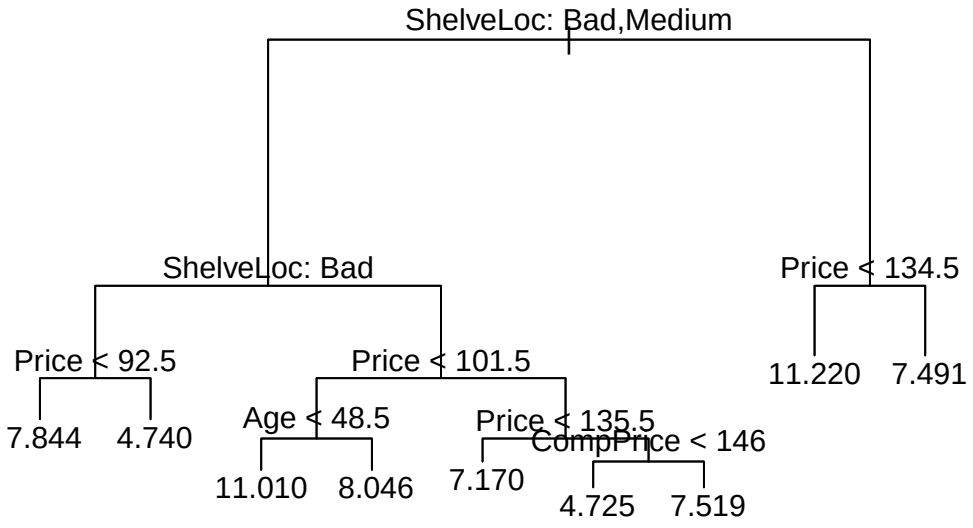
```
##
## Regression tree:
## tree(formula = Sales ~ ., data = Carseats.train)
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "Age"         "CompPrice"   "Advertising"
## [6] "US"
## Number of terminal nodes:  16
## Residual mean deviance:  2.134 = 392.6 / 184
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -4.37400 -0.90790 -0.05181  0.00000  0.92840  3.82600
```




```
## [1] 4.784151
```

(C) In this case, the tree of size 8 is selected by cross-validation. Pruning the tree increases the MSE.
MSE: 5.075903
Functions: cv.tree(), prune.tree()


cv.carseats$dev
1000
1200
1400
5
10
15
cv.carseats$size


ShelveLoc: Bad,Medium
ShelveLoc: Bad
Price < 134.5
Price < 92.5
Price < 101.5
11.220
7.491
7.844
4.740
Age < 48.5
Price < 135.5
CompPrice < 146
11.010
8.046
7.170
4.725
7.519

```
## [1] 5.075903
```

(d) Bagging improves the test MSE to 2.79. We also see that Price, ShelveLoc and Age are three most important predictors of Sale.
MSE: 2.795264
Functions: randomForest(), importance()

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
## [1] 2.758793
```

```
##               %IncMSE IncNodePurity
## CompPrice   20.724998    130.421567
## Income       2.616103     66.153373
## Advertising 14.214948    121.847519
## Population  -1.433690     58.523885
## Price       50.544432    408.079671
## ShelveLoc   57.131042    495.464946
## Age         14.442394    118.497755
## Education   -1.849036     38.905898
## Urban       -3.593040      7.957335
## US           5.850580     11.115617
```

(e) Random forest worsen the test MSE to 3.4. We again see that Price and ShelveLoc are two most important predictors of Sale.
MSE: 3.400033
Functions: randomForest(), importance()

```
## [1] 3.36742
```

```
##                 %IncMSE IncNodePurity
## CompPrice    9.13206555     127.87589
## Income       0.55571004     106.44280
## Advertising 13.42575151     170.45304
## Population  -0.09152891      97.55192
## Price       31.14563456     324.35874
## ShelveLoc   37.35563450     348.94231
## Age          8.85914380     131.46274
## Education   -0.48270002      61.54373
## Urban        0.48953546      14.00106
## US           7.55566167      33.73913
```

Chapter 8: #11

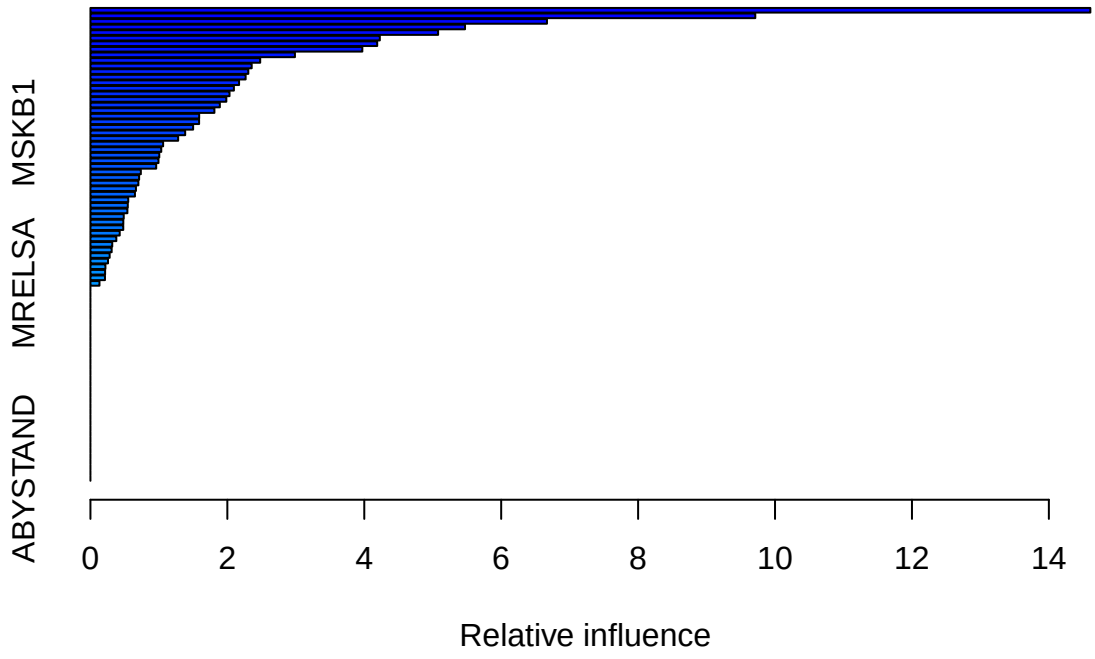
(a) Caravan$Purchase = ifelse(Caravan$Purchase == \text{“Yes”}, 1, 0)$

```
## Loaded gbm 2.1.5
```

(b) The variables “PPERSAUT” and “MKOOPKLA” are the two most important variables.
Functions: gbm()

```
## Warning in gbm.fit(x = x, y = y, offset = offset, distribution =
## distribution, : variable 50: PVRAAUT has no variation.

## Warning in gbm.fit(x = x, y = y, offset = offset, distribution =
## distribution, : variable 71: AVRAAUT has no variation.
```




```
##                var    rel.inf
## PPERSAUT PPERSAUT 14.6059360
## MKOOPKLA MKOOPKLA  9.7117578
## MOPLHOOG MOPLHOOG  6.6684660
## MBERMIDD MBERMIDD  5.4718168
## PBRAND     PBRAND  5.0787300
## ABRAND     ABRAND  4.2271757
## MGODGE     MGODGE  4.1891158
## MINK3045 MINK3045  3.9715766
## MSKA         MSKA  2.9870647
## PWAPART   PWAPART  2.4795405
## MAUT1       MAUT1  2.3553134
## MOSTYPE   MOSTYPE  2.3063541
## MSKC         MSKC  2.2671449
## MGODOV     MGODOV  2.1712568
## MAUT2       MAUT2  2.0956056
## MGODPR     MGODPR  2.0318958
## MBERARBG MBERARBG  1.9851925
## MBERHOOG MBERHOOG  1.8903304
```

```
## MFWEKIND MFWEKIND  1.8108823
## MINKGEM   MINKGEM  1.5886284
## MINK7512 MINK7512  1.5872351
## PBYSTAND PBYSTAND  1.5000945
## MSKB1       MSKB1  1.3841040
## MRELGE     MRELGE  1.2814393
## MFGEKIND MFGEKIND  1.0617962
## APERSAUT APERSAUT  1.0345402
## MRELOV     MRELOV  1.0056086
## MSKD         MSKD  0.9939787
## MGODRK     MGODRK  0.9606337
## MOPLMIDD MOPLMIDD  0.7360710
## MHKOOP     MHKOOP  0.7110800
## MHHUUR     MHHUUR  0.7005076
## MAUT0       MAUT0  0.6647820
## MSKB2       MSKB2  0.6499879
## PLEVEN     PLEVEN  0.5524622
## MZFONDS   MZFONDS  0.5458052
## MBERBOER MBERBOER  0.5398413
## MINKM30   MINKM30  0.4865484
## MOSHOOFD MOSHOOFD  0.4804510
## MGEMOMV   MGEMOMV  0.4794458
## MINK4575 MINK4575  0.4275559
## PMOTSCO   PMOTSCO  0.3781598
## MOPLLAAG MOPLLAAG  0.3185780
## MBERARBO MBERARBO  0.3101480
## MZPART     MZPART  0.2809377
## MGEMLEEF MGEMLEEF  0.2565490
## MINK123M MINK123M  0.2189397
## MBERZELF MBERZELF  0.2143890
## MFALLEEN MFALLEEN  0.2123572
## MRELSA     MRELSA  0.1321887
## MAANTHUI MAANTHUI  0.0000000
## PWABEDR   PWABEDR  0.0000000
## PWALAND   PWALAND  0.0000000
## PBESAUT   PBESAUT  0.0000000
## PVRAAUT   PVRAAUT  0.0000000
## PAANHANG PAANHANG  0.0000000
## PTRACTOR PTRACTOR  0.0000000
## PWERKT     PWERKT  0.0000000
## PBROM       PBROM  0.0000000
## PPERSONG PPERSONG  0.0000000
## PGEZONG   PGEZONG  0.0000000
## PWAOREG   PWAOREG  0.0000000
## PZEILPL   PZEILPL  0.0000000
## PPLEZIER PPLEZIER  0.0000000
## PFIETS     PFIETS  0.0000000
## PINBOED   PINBOED  0.0000000
## AWAPART   AWAPART  0.0000000
## AWABEDR   AWABEDR  0.0000000
## AWALAND   AWALAND  0.0000000
## ABESAUT   ABESAUT  0.0000000
## AMOTSCO   AMOTSCO  0.0000000
## AVRAAUT   AVRAAUT  0.0000000
```

```
## AAANHANG AAANHANG  0.0000000
## ATRACTOR ATRACTOR  0.0000000
## AWERKT     AWERKT  0.0000000
## ABROM       ABROM  0.0000000
## ALEVEN     ALEVEN  0.0000000
## APERSONG APERSONG  0.0000000
## AGEZONG   AGEZONG  0.0000000
## AWAOREG   AWAOREG  0.0000000
## AZEILPL   AZEILPL  0.0000000
## APLEZIER APLEZIER  0.0000000
## AFIETS     AFIETS  0.0000000
## AINBOED   AINBOED  0.0000000
## ABYSTAND ABYSTAND  0.0000000
```

(C) About 22.07% of people predicted to make purchase actually end up making one. For logistic regression, the fraction of people predicted to make a purchase that in fact make one is again 22.07%.

```
## [1] "Boosting"
```

```
##    pred.test
##        0    1
##   0 4413  120
##   1  255   34
```

```
## [1] 0.2207792
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
## == : prediction from a rank-deficient fit may be misleading
```

```
## [1] "Logistic Regression"
```

```
##    pred.test2
##        0    1
##   0 4413  120
##   1  255   34
```

```
## [1] 0.2207792
```

Problem 1: Beauty Pays!

Problem 4: BART

Problem 5: Neural Nets