Predictive Modeling Take Home Exam

Sujay Chebbi

08/02/2020

Exam questions

Chapter 2 - 10

Chapter 2 - 10a

```r
rm(list = ls())
library(MASS)
Boston
```

```
##         crim   zn indus chas    nox    rm   age     dis rad tax ptratio  black
## 1    0.00632 18.0  2.31    0 0.5380 6.575  65.2  4.0900   1 296    15.3 396.90
## 2    0.02731  0.0  7.07    0 0.4690 6.421  78.9  4.9671   2 242    17.8 396.90
## 3    0.02729  0.0  7.07    0 0.4690 7.185  61.1  4.9671   2 242    17.8 392.83
## 4    0.03237  0.0  2.18    0 0.4580 6.998  45.8  6.0622   3 222    18.7 394.63
## 5    0.06905  0.0  2.18    0 0.4580 7.147  54.2  6.0622   3 222    18.7 396.90
## 6    0.02985  0.0  2.18    0 0.4580 6.430  58.7  6.0622   3 222    18.7 394.12
## 7    0.08829 12.5  7.87    0 0.5240 6.012  66.6  5.5605   5 311    15.2 395.60
## 8    0.14455 12.5  7.87    0 0.5240 6.172  96.1  5.9505   5 311    15.2 396.90
## 9    0.21124 12.5  7.87    0 0.5240 5.631 100.0  6.0821   5 311    15.2 386.63
## 10   0.17004 12.5  7.87    0 0.5240 6.004  85.9  6.5921   5 311    15.2 386.71
## 11   0.22489 12.5  7.87    0 0.5240 6.377  94.3  6.3467   5 311    15.2 392.52
## 12   0.11747 12.5  7.87    0 0.5240 6.009  82.9  6.2267   5 311    15.2 396.90
## 13   0.09378 12.5  7.87    0 0.5240 5.889  39.0  5.4509   5 311    15.2 390.50
## 14   0.62976  0.0  8.14    0 0.5380 5.949  61.8  4.7075   4 307    21.0 396.90
## 15   0.63796  0.0  8.14    0 0.5380 6.096  84.5  4.4619   4 307    21.0 380.02
## 16   0.62739  0.0  8.14    0 0.5380 5.834  56.5  4.4986   4 307    21.0 395.62
## 17   1.05393  0.0  8.14    0 0.5380 5.935  29.3  4.4986   4 307    21.0 386.85
## 18   0.78420  0.0  8.14    0 0.5380 5.990  81.7  4.2579   4 307    21.0 386.75
## 19   0.80271  0.0  8.14    0 0.5380 5.456  36.6  3.7965   4 307    21.0 288.99
## 20   0.72580  0.0  8.14    0 0.5380 5.727  69.5  3.7965   4 307    21.0 390.95
## 21   1.25179  0.0  8.14    0 0.5380 5.570  98.1  3.7979   4 307    21.0 376.57
## 22   0.85204  0.0  8.14    0 0.5380 5.965  89.2  4.0123   4 307    21.0 392.53
## 23   1.23247  0.0  8.14    0 0.5380 6.142  91.7  3.9769   4 307    21.0 396.90
## 24   0.98843  0.0  8.14    0 0.5380 5.813 100.0  4.0952   4 307    21.0 394.54
## 25   0.75026  0.0  8.14    0 0.5380 5.924  94.1  4.3996   4 307    21.0 394.33
## 26   0.84054  0.0  8.14    0 0.5380 5.599  85.7  4.4546   4 307    21.0 303.42
## 27   0.67191  0.0  8.14    0 0.5380 5.813  90.3  4.6820   4 307    21.0 376.88
## 28   0.95577  0.0  8.14    0 0.5380 6.047  88.8  4.4534   4 307    21.0 306.38
## 29   0.77299  0.0  8.14    0 0.5380 6.495  94.4  4.4547   4 307    21.0 387.94
## 30   1.00245  0.0  8.14    0 0.5380 6.674  87.3  4.2390   4 307    21.0 380.23
## 31   1.13081  0.0  8.14    0 0.5380 5.713  94.1  4.2330   4 307    21.0 360.17
## 32   1.35472  0.0  8.14    0 0.5380 6.072 100.0  4.1750   4 307    21.0 376.73
```

```
## 33   1.38799   0.0  8.14    0 0.5380 5.950  82.0  3.9900   4 307     21.0 232.60
## 34   1.15172   0.0  8.14    0 0.5380 5.701  95.0  3.7872   4 307     21.0 358.77
## 35   1.61282   0.0  8.14    0 0.5380 6.096  96.9  3.7598   4 307     21.0 248.31
## 36   0.06417   0.0  5.96    0 0.4990 5.933  68.2  3.3603   5 279     19.2 396.90
## 37   0.09744   0.0  5.96    0 0.4990 5.841  61.4  3.3779   5 279     19.2 377.56
## 38   0.08014   0.0  5.96    0 0.4990 5.850  41.5  3.9342   5 279     19.2 396.90
## 39   0.17505   0.0  5.96    0 0.4990 5.966  30.2  3.8473   5 279     19.2 393.43
## 40   0.02763  75.0  2.95    0 0.4280 6.595  21.8  5.4011   3 252     18.3 395.63
## 41   0.03359  75.0  2.95    0 0.4280 7.024  15.8  5.4011   3 252     18.3 395.62
## 42   0.12744   0.0  6.91    0 0.4480 6.770   2.9  5.7209   3 233     17.9 385.41
## 43   0.14150   0.0  6.91    0 0.4480 6.169   6.6  5.7209   3 233     17.9 383.37
## 44   0.15936   0.0  6.91    0 0.4480 6.211   6.5  5.7209   3 233     17.9 394.46
## 45   0.12269   0.0  6.91    0 0.4480 6.069  40.0  5.7209   3 233     17.9 389.39
## 46   0.17142   0.0  6.91    0 0.4480 5.682  33.8  5.1004   3 233     17.9 396.90
## 47   0.18836   0.0  6.91    0 0.4480 5.786  33.3  5.1004   3 233     17.9 396.90
## 48   0.22927   0.0  6.91    0 0.4480 6.030  85.5  5.6894   3 233     17.9 392.74
## 49   0.25387   0.0  6.91    0 0.4480 5.399  95.3  5.8700   3 233     17.9 396.90
## 50   0.21977   0.0  6.91    0 0.4480 5.602  62.0  6.0877   3 233     17.9 396.90
## 51   0.08873  21.0  5.64    0 0.4390 5.963  45.7  6.8147   4 243     16.8 395.56
## 52   0.04337  21.0  5.64    0 0.4390 6.115  63.0  6.8147   4 243     16.8 393.97
## 53   0.05360  21.0  5.64    0 0.4390 6.511  21.1  6.8147   4 243     16.8 396.90
## 54   0.04981  21.0  5.64    0 0.4390 5.998  21.4  6.8147   4 243     16.8 396.90
## 55   0.01360  75.0  4.00    0 0.4100 5.888  47.6  7.3197   3 469     21.1 396.90
## 56   0.01311  90.0  1.22    0 0.4030 7.249  21.9  8.6966   5 226     17.9 395.93
## 57   0.02055  85.0  0.74    0 0.4100 6.383  35.7  9.1876   2 313     17.3 396.90
## 58   0.01432 100.0  1.32    0 0.4110 6.816  40.5  8.3248   5 256     15.1 392.90
## 59   0.15445  25.0  5.13    0 0.4530 6.145  29.2  7.8148   8 284     19.7 390.68
## 60   0.10328  25.0  5.13    0 0.4530 5.927  47.2  6.9320   8 284     19.7 396.90
## 61   0.14932  25.0  5.13    0 0.4530 5.741  66.2  7.2254   8 284     19.7 395.11
## 62   0.17171  25.0  5.13    0 0.4530 5.966  93.4  6.8185   8 284     19.7 378.08
## 63   0.11027  25.0  5.13    0 0.4530 6.456  67.8  7.2255   8 284     19.7 396.90
## 64   0.12650  25.0  5.13    0 0.4530 6.762  43.4  7.9809   8 284     19.7 395.58
## 65   0.01951  17.5  1.38    0 0.4161 7.104  59.5  9.2229   3 216     18.6 393.24
## 66   0.03584  80.0  3.37    0 0.3980 6.290  17.8  6.6115   4 337     16.1 396.90
## 67   0.04379  80.0  3.37    0 0.3980 5.787  31.1  6.6115   4 337     16.1 396.90
## 68   0.05789  12.5  6.07    0 0.4090 5.878  21.4  6.4980   4 345     18.9 396.21
## 69   0.13554  12.5  6.07    0 0.4090 5.594  36.8  6.4980   4 345     18.9 396.90
## 70   0.12816  12.5  6.07    0 0.4090 5.885  33.0  6.4980   4 345     18.9 396.90
## 71   0.08826   0.0 10.81    0 0.4130 6.417   6.6  5.2873   4 305     19.2 383.73
## 72   0.15876   0.0 10.81    0 0.4130 5.961  17.5  5.2873   4 305     19.2 376.94
## 73   0.09164   0.0 10.81    0 0.4130 6.065   7.8  5.2873   4 305     19.2 390.91
## 74   0.19539   0.0 10.81    0 0.4130 6.245   6.2  5.2873   4 305     19.2 377.17
## 75   0.07896   0.0 12.83    0 0.4370 6.273   6.0  4.2515   5 398     18.7 394.92
## 76   0.09512   0.0 12.83    0 0.4370 6.286  45.0  4.5026   5 398     18.7 383.23
## 77   0.10153   0.0 12.83    0 0.4370 6.279  74.5  4.0522   5 398     18.7 373.66
## 78   0.08707   0.0 12.83    0 0.4370 6.140  45.8  4.0905   5 398     18.7 386.96
## 79   0.05646   0.0 12.83    0 0.4370 6.232  53.7  5.0141   5 398     18.7 386.40
## 80   0.08387   0.0 12.83    0 0.4370 5.874  36.6  4.5026   5 398     18.7 396.06
## 81   0.04113  25.0  4.86    0 0.4260 6.727  33.5  5.4007   4 281     19.0 396.90
## 82   0.04462  25.0  4.86    0 0.4260 6.619  70.4  5.4007   4 281     19.0 395.63
## 83   0.03659  25.0  4.86    0 0.4260 6.302  32.2  5.4007   4 281     19.0 396.90
## 84   0.03551  25.0  4.86    0 0.4260 6.167  46.7  5.4007   4 281     19.0 390.64
## 85   0.05059   0.0  4.49    0 0.4490 6.389  48.0  4.7794   3 247     18.5 396.90
## 86   0.05735   0.0  4.49    0 0.4490 6.630  56.1  4.4377   3 247     18.5 392.30
```

```
## 87   0.05188  0.0  4.49    0 0.4490 6.015  45.1  4.4272   3 247    18.5 395.99
## 88   0.07151  0.0  4.49    0 0.4490 6.121  56.8  3.7476   3 247    18.5 395.15
## 89   0.05660  0.0  3.41    0 0.4890 7.007  86.3  3.4217   2 270    17.8 396.90
## 90   0.05302  0.0  3.41    0 0.4890 7.079  63.1  3.4145   2 270    17.8 396.06
## 91   0.04684  0.0  3.41    0 0.4890 6.417  66.1  3.0923   2 270    17.8 392.18
## 92   0.03932  0.0  3.41    0 0.4890 6.405  73.9  3.0921   2 270    17.8 393.55
## 93   0.04203 28.0 15.04    0 0.4640 6.442  53.6  3.6659   4 270    18.2 395.01
## 94   0.02875 28.0 15.04    0 0.4640 6.211  28.9  3.6659   4 270    18.2 396.33
## 95   0.04294 28.0 15.04    0 0.4640 6.249  77.3  3.6150   4 270    18.2 396.90
## 96   0.12204  0.0  2.89    0 0.4450 6.625  57.8  3.4952   2 276    18.0 357.98
## 97   0.11504  0.0  2.89    0 0.4450 6.163  69.6  3.4952   2 276    18.0 391.83
## 98   0.12083  0.0  2.89    0 0.4450 8.069  76.0  3.4952   2 276    18.0 396.90
## 99   0.08187  0.0  2.89    0 0.4450 7.820  36.9  3.4952   2 276    18.0 393.53
## 100  0.06860  0.0  2.89    0 0.4450 7.416  62.5  3.4952   2 276    18.0 396.90
## 101  0.14866  0.0  8.56    0 0.5200 6.727  79.9  2.7778   5 384    20.9 394.76
## 102  0.11432  0.0  8.56    0 0.5200 6.781  71.3  2.8561   5 384    20.9 395.58
## 103  0.22876  0.0  8.56    0 0.5200 6.405  85.4  2.7147   5 384    20.9  70.80
## 104  0.21161  0.0  8.56    0 0.5200 6.137  87.4  2.7147   5 384    20.9 394.47
## 105  0.13960  0.0  8.56    0 0.5200 6.167  90.0  2.4210   5 384    20.9 392.69
## 106  0.13262  0.0  8.56    0 0.5200 5.851  96.7  2.1069   5 384    20.9 394.05
## 107  0.17120  0.0  8.56    0 0.5200 5.836  91.9  2.2110   5 384    20.9 395.67
## 108  0.13117  0.0  8.56    0 0.5200 6.127  85.2  2.1224   5 384    20.9 387.69
## 109  0.12802  0.0  8.56    0 0.5200 6.474  97.1  2.4329   5 384    20.9 395.24
## 110  0.26363  0.0  8.56    0 0.5200 6.229  91.2  2.5451   5 384    20.9 391.23
## 111  0.10793  0.0  8.56    0 0.5200 6.195  54.4  2.7778   5 384    20.9 393.49
## 112  0.10084  0.0 10.01    0 0.5470 6.715  81.6  2.6775   6 432    17.8 395.59
## 113  0.12329  0.0 10.01    0 0.5470 5.913  92.9  2.3534   6 432    17.8 394.95
## 114  0.22212  0.0 10.01    0 0.5470 6.092  95.4  2.5480   6 432    17.8 396.90
## 115  0.14231  0.0 10.01    0 0.5470 6.254  84.2  2.2565   6 432    17.8 388.74
## 116  0.17134  0.0 10.01    0 0.5470 5.928  88.2  2.4631   6 432    17.8 344.91
## 117  0.13158  0.0 10.01    0 0.5470 6.176  72.5  2.7301   6 432    17.8 393.30
## 118  0.15098  0.0 10.01    0 0.5470 6.021  82.6  2.7474   6 432    17.8 394.51
## 119  0.13058  0.0 10.01    0 0.5470 5.872  73.1  2.4775   6 432    17.8 338.63
## 120  0.14476  0.0 10.01    0 0.5470 5.731  65.2  2.7592   6 432    17.8 391.50
## 121  0.06899  0.0 25.65    0 0.5810 5.870  69.7  2.2577   2 188    19.1 389.15
## 122  0.07165  0.0 25.65    0 0.5810 6.004  84.1  2.1974   2 188    19.1 377.67
## 123  0.09299  0.0 25.65    0 0.5810 5.961  92.9  2.0869   2 188    19.1 378.09
## 124  0.15038  0.0 25.65    0 0.5810 5.856  97.0  1.9444   2 188    19.1 370.31
## 125  0.09849  0.0 25.65    0 0.5810 5.879  95.8  2.0063   2 188    19.1 379.38
## 126  0.16902  0.0 25.65    0 0.5810 5.986  88.4  1.9929   2 188    19.1 385.02
## 127  0.38735  0.0 25.65    0 0.5810 5.613  95.6  1.7572   2 188    19.1 359.29
## 128  0.25915  0.0 21.89    0 0.6240 5.693  96.0  1.7883   4 437    21.2 392.11
## 129  0.32543  0.0 21.89    0 0.6240 6.431  98.8  1.8125   4 437    21.2 396.90
## 130  0.88125  0.0 21.89    0 0.6240 5.637  94.7  1.9799   4 437    21.2 396.90
## 131  0.34006  0.0 21.89    0 0.6240 6.458  98.9  2.1185   4 437    21.2 395.04
## 132  1.19294  0.0 21.89    0 0.6240 6.326  97.7  2.2710   4 437    21.2 396.90
## 133  0.59005  0.0 21.89    0 0.6240 6.372  97.9  2.3274   4 437    21.2 385.76
## 134  0.32982  0.0 21.89    0 0.6240 5.822  95.4  2.4699   4 437    21.2 388.69
## 135  0.97617  0.0 21.89    0 0.6240 5.757  98.4  2.3460   4 437    21.2 262.76
## 136  0.55778  0.0 21.89    0 0.6240 6.335  98.2  2.1107   4 437    21.2 394.67
## 137  0.32264  0.0 21.89    0 0.6240 5.942  93.5  1.9669   4 437    21.2 378.25
## 138  0.35233  0.0 21.89    0 0.6240 6.454  98.4  1.8498   4 437    21.2 394.08
## 139  0.24980  0.0 21.89    0 0.6240 5.857  98.2  1.6686   4 437    21.2 392.04
## 140  0.54452  0.0 21.89    0 0.6240 6.151  97.9  1.6687   4 437    21.2 396.90
```

```
## 141  0.29090   0.0 21.89    0 0.6240 6.174  93.6  1.6119   4 437    21.2 388.08
## 142  1.62864   0.0 21.89    0 0.6240 5.019 100.0  1.4394   4 437    21.2 396.90
## 143  3.32105   0.0 19.58    1 0.8710 5.403 100.0  1.3216   5 403    14.7 396.90
## 144  4.09740   0.0 19.58    0 0.8710 5.468 100.0  1.4118   5 403    14.7 396.90
## 145  2.77974   0.0 19.58    0 0.8710 4.903  97.8  1.3459   5 403    14.7 396.90
## 146  2.37934   0.0 19.58    0 0.8710 6.130 100.0  1.4191   5 403    14.7 172.91
## 147  2.15505   0.0 19.58    0 0.8710 5.628 100.0  1.5166   5 403    14.7 169.27
## 148  2.36862   0.0 19.58    0 0.8710 4.926  95.7  1.4608   5 403    14.7 391.71
## 149  2.33099   0.0 19.58    0 0.8710 5.186  93.8  1.5296   5 403    14.7 356.99
## 150  2.73397   0.0 19.58    0 0.8710 5.597  94.9  1.5257   5 403    14.7 351.85
## 151  1.65660   0.0 19.58    0 0.8710 6.122  97.3  1.6180   5 403    14.7 372.80
## 152  1.49632   0.0 19.58    0 0.8710 5.404 100.0  1.5916   5 403    14.7 341.60
## 153  1.12658   0.0 19.58    1 0.8710 5.012  88.0  1.6102   5 403    14.7 343.28
## 154  2.14918   0.0 19.58    0 0.8710 5.709  98.5  1.6232   5 403    14.7 261.95
## 155  1.41385   0.0 19.58    1 0.8710 6.129  96.0  1.7494   5 403    14.7 321.02
## 156  3.53501   0.0 19.58    1 0.8710 6.152  82.6  1.7455   5 403    14.7  88.01
## 157  2.44668   0.0 19.58    0 0.8710 5.272  94.0  1.7364   5 403    14.7  88.63
## 158  1.22358   0.0 19.58    0 0.6050 6.943  97.4  1.8773   5 403    14.7 363.43
## 159  1.34284   0.0 19.58    0 0.6050 6.066 100.0  1.7573   5 403    14.7 353.89
## 160  1.42502   0.0 19.58    0 0.8710 6.510 100.0  1.7659   5 403    14.7 364.31
## 161  1.27346   0.0 19.58    1 0.6050 6.250  92.6  1.7984   5 403    14.7 338.92
## 162  1.46336   0.0 19.58    0 0.6050 7.489  90.8  1.9709   5 403    14.7 374.43
## 163  1.83377   0.0 19.58    1 0.6050 7.802  98.2  2.0407   5 403    14.7 389.61
## 164  1.51902   0.0 19.58    1 0.6050 8.375  93.9  2.1620   5 403    14.7 388.45
## 165  2.24236   0.0 19.58    0 0.6050 5.854  91.8  2.4220   5 403    14.7 395.11
## 166  2.92400   0.0 19.58    0 0.6050 6.101  93.0  2.2834   5 403    14.7 240.16
## 167  2.01019   0.0 19.58    0 0.6050 7.929  96.2  2.0459   5 403    14.7 369.30
## 168  1.80028   0.0 19.58    0 0.6050 5.877  79.2  2.4259   5 403    14.7 227.61
## 169  2.30040   0.0 19.58    0 0.6050 6.319  96.1  2.1000   5 403    14.7 297.09
## 170  2.44953   0.0 19.58    0 0.6050 6.402  95.2  2.2625   5 403    14.7 330.04
## 171  1.20742   0.0 19.58    0 0.6050 5.875  94.6  2.4259   5 403    14.7 292.29
## 172  2.31390   0.0 19.58    0 0.6050 5.880  97.3  2.3887   5 403    14.7 348.13
## 173  0.13914   0.0  4.05    0 0.5100 5.572  88.5  2.5961   5 296    16.6 396.90
## 174  0.09178   0.0  4.05    0 0.5100 6.416  84.1  2.6463   5 296    16.6 395.50
## 175  0.08447   0.0  4.05    0 0.5100 5.859  68.7  2.7019   5 296    16.6 393.23
## 176  0.06664   0.0  4.05    0 0.5100 6.546  33.1  3.1323   5 296    16.6 390.96
## 177  0.07022   0.0  4.05    0 0.5100 6.020  47.2  3.5549   5 296    16.6 393.23
## 178  0.05425   0.0  4.05    0 0.5100 6.315  73.4  3.3175   5 296    16.6 395.60
## 179  0.06642   0.0  4.05    0 0.5100 6.860  74.4  2.9153   5 296    16.6 391.27
## 180  0.05780   0.0  2.46    0 0.4880 6.980  58.4  2.8290   3 193    17.8 396.90
## 181  0.06588   0.0  2.46    0 0.4880 7.765  83.3  2.7410   3 193    17.8 395.56
## 182  0.06888   0.0  2.46    0 0.4880 6.144  62.2  2.5979   3 193    17.8 396.90
## 183  0.09103   0.0  2.46    0 0.4880 7.155  92.2  2.7006   3 193    17.8 394.12
## 184  0.10008   0.0  2.46    0 0.4880 6.563  95.6  2.8470   3 193    17.8 396.90
## 185  0.08308   0.0  2.46    0 0.4880 5.604  89.8  2.9879   3 193    17.8 391.00
## 186  0.06047   0.0  2.46    0 0.4880 6.153  68.8  3.2797   3 193    17.8 387.11
## 187  0.05602   0.0  2.46    0 0.4880 7.831  53.6  3.1992   3 193    17.8 392.63
## 188  0.07875  45.0  3.44    0 0.4370 6.782  41.1  3.7886   5 398    15.2 393.87
## 189  0.12579  45.0  3.44    0 0.4370 6.556  29.1  4.5667   5 398    15.2 382.84
## 190  0.08370  45.0  3.44    0 0.4370 7.185  38.9  4.5667   5 398    15.2 396.90
## 191  0.09068  45.0  3.44    0 0.4370 6.951  21.5  6.4798   5 398    15.2 377.68
## 192  0.06911  45.0  3.44    0 0.4370 6.739  30.8  6.4798   5 398    15.2 389.71
## 193  0.08664  45.0  3.44    0 0.4370 7.178  26.3  6.4798   5 398    15.2 390.49
## 194  0.02187  60.0  2.93    0 0.4010 6.800   9.9  6.2196   1 265    15.6 393.37
```

```
## 195  0.01439  60.0  2.93    0 0.4010 6.604  18.8  6.2196   1 265     15.6 376.70
## 196  0.01381  80.0  0.46    0 0.4220 7.875  32.0  5.6484   4 255     14.4 394.23
## 197  0.04011  80.0  1.52    0 0.4040 7.287  34.1  7.3090   2 329     12.6 396.90
## 198  0.04666  80.0  1.52    0 0.4040 7.107  36.6  7.3090   2 329     12.6 354.31
## 199  0.03768  80.0  1.52    0 0.4040 7.274  38.3  7.3090   2 329     12.6 392.20
## 200  0.03150  95.0  1.47    0 0.4030 6.975  15.3  7.6534   3 402     17.0 396.90
## 201  0.01778  95.0  1.47    0 0.4030 7.135  13.9  7.6534   3 402     17.0 384.30
## 202  0.03445  82.5  2.03    0 0.4150 6.162  38.4  6.2700   2 348     14.7 393.77
## 203  0.02177  82.5  2.03    0 0.4150 7.610  15.7  6.2700   2 348     14.7 395.38
## 204  0.03510  95.0  2.68    0 0.4161 7.853  33.2  5.1180   4 224     14.7 392.78
## 205  0.02009  95.0  2.68    0 0.4161 8.034  31.9  5.1180   4 224     14.7 390.55
## 206  0.13642   0.0 10.59    0 0.4890 5.891  22.3  3.9454   4 277     18.6 396.90
## 207  0.22969   0.0 10.59    0 0.4890 6.326  52.5  4.3549   4 277     18.6 394.87
## 208  0.25199   0.0 10.59    0 0.4890 5.783  72.7  4.3549   4 277     18.6 389.43
## 209  0.13587   0.0 10.59    1 0.4890 6.064  59.1  4.2392   4 277     18.6 381.32
## 210  0.43571   0.0 10.59    1 0.4890 5.344 100.0  3.8750   4 277     18.6 396.90
## 211  0.17446   0.0 10.59    1 0.4890 5.960  92.1  3.8771   4 277     18.6 393.25
## 212  0.37578   0.0 10.59    1 0.4890 5.404  88.6  3.6650   4 277     18.6 395.24
## 213  0.21719   0.0 10.59    1 0.4890 5.807  53.8  3.6526   4 277     18.6 390.94
## 214  0.14052   0.0 10.59    0 0.4890 6.375  32.3  3.9454   4 277     18.6 385.81
## 215  0.28955   0.0 10.59    0 0.4890 5.412   9.8  3.5875   4 277     18.6 348.93
## 216  0.19802   0.0 10.59    0 0.4890 6.182  42.4  3.9454   4 277     18.6 393.63
## 217  0.04560   0.0 13.89    1 0.5500 5.888  56.0  3.1121   5 276     16.4 392.80
## 218  0.07013   0.0 13.89    0 0.5500 6.642  85.1  3.4211   5 276     16.4 392.78
## 219  0.11069   0.0 13.89    1 0.5500 5.951  93.8  2.8893   5 276     16.4 396.90
## 220  0.11425   0.0 13.89    1 0.5500 6.373  92.4  3.3633   5 276     16.4 393.74
## 221  0.35809   0.0  6.20    1 0.5070 6.951  88.5  2.8617   8 307     17.4 391.70
## 222  0.40771   0.0  6.20    1 0.5070 6.164  91.3  3.0480   8 307     17.4 395.24
## 223  0.62356   0.0  6.20    1 0.5070 6.879  77.7  3.2721   8 307     17.4 390.39
## 224  0.61470   0.0  6.20    0 0.5070 6.618  80.8  3.2721   8 307     17.4 396.90
## 225  0.31533   0.0  6.20    0 0.5040 8.266  78.3  2.8944   8 307     17.4 385.05
## 226  0.52693   0.0  6.20    0 0.5040 8.725  83.0  2.8944   8 307     17.4 382.00
## 227  0.38214   0.0  6.20    0 0.5040 8.040  86.5  3.2157   8 307     17.4 387.38
## 228  0.41238   0.0  6.20    0 0.5040 7.163  79.9  3.2157   8 307     17.4 372.08
## 229  0.29819   0.0  6.20    0 0.5040 7.686  17.0  3.3751   8 307     17.4 377.51
## 230  0.44178   0.0  6.20    0 0.5040 6.552  21.4  3.3751   8 307     17.4 380.34
## 231  0.53700   0.0  6.20    0 0.5040 5.981  68.1  3.6715   8 307     17.4 378.35
## 232  0.46296   0.0  6.20    0 0.5040 7.412  76.9  3.6715   8 307     17.4 376.14
## 233  0.57529   0.0  6.20    0 0.5070 8.337  73.3  3.8384   8 307     17.4 385.91
## 234  0.33147   0.0  6.20    0 0.5070 8.247  70.4  3.6519   8 307     17.4 378.95
## 235  0.44791   0.0  6.20    1 0.5070 6.726  66.5  3.6519   8 307     17.4 360.20
## 236  0.33045   0.0  6.20    0 0.5070 6.086  61.5  3.6519   8 307     17.4 376.75
## 237  0.52058   0.0  6.20    1 0.5070 6.631  76.5  4.1480   8 307     17.4 388.45
## 238  0.51183   0.0  6.20    0 0.5070 7.358  71.6  4.1480   8 307     17.4 390.07
## 239  0.08244  30.0  4.93    0 0.4280 6.481  18.5  6.1899   6 300     16.6 379.41
## 240  0.09252  30.0  4.93    0 0.4280 6.606  42.2  6.1899   6 300     16.6 383.78
## 241  0.11329  30.0  4.93    0 0.4280 6.897  54.3  6.3361   6 300     16.6 391.25
## 242  0.10612  30.0  4.93    0 0.4280 6.095  65.1  6.3361   6 300     16.6 394.62
## 243  0.10290  30.0  4.93    0 0.4280 6.358  52.9  7.0355   6 300     16.6 372.75
## 244  0.12757  30.0  4.93    0 0.4280 6.393   7.8  7.0355   6 300     16.6 374.71
## 245  0.20608  22.0  5.86    0 0.4310 5.593  76.5  7.9549   7 330     19.1 372.49
## 246  0.19133  22.0  5.86    0 0.4310 5.605  70.2  7.9549   7 330     19.1 389.13
## 247  0.33983  22.0  5.86    0 0.4310 6.108  34.9  8.0555   7 330     19.1 390.18
## 248  0.19657  22.0  5.86    0 0.4310 6.226  79.2  8.0555   7 330     19.1 376.14
```

```
## 249  0.16439  22.0  5.86    0 0.4310 6.433  49.1  7.8265   7 330    19.1 374.71
## 250  0.19073  22.0  5.86    0 0.4310 6.718  17.5  7.8265   7 330    19.1 393.74
## 251  0.14030  22.0  5.86    0 0.4310 6.487  13.0  7.3967   7 330    19.1 396.28
## 252  0.21409  22.0  5.86    0 0.4310 6.438   8.9  7.3967   7 330    19.1 377.07
## 253  0.08221  22.0  5.86    0 0.4310 6.957   6.8  8.9067   7 330    19.1 386.09
## 254  0.36894  22.0  5.86    0 0.4310 8.259   8.4  8.9067   7 330    19.1 396.90
## 255  0.04819  80.0  3.64    0 0.3920 6.108  32.0  9.2203   1 315    16.4 392.89
## 256  0.03548  80.0  3.64    0 0.3920 5.876  19.1  9.2203   1 315    16.4 395.18
## 257  0.01538  90.0  3.75    0 0.3940 7.454  34.2  6.3361   3 244    15.9 386.34
## 258  0.61154  20.0  3.97    0 0.6470 8.704  86.9  1.8010   5 264    13.0 389.70
## 259  0.66351  20.0  3.97    0 0.6470 7.333 100.0  1.8946   5 264    13.0 383.29
## 260  0.65665  20.0  3.97    0 0.6470 6.842 100.0  2.0107   5 264    13.0 391.93
## 261  0.54011  20.0  3.97    0 0.6470 7.203  81.8  2.1121   5 264    13.0 392.80
## 262  0.53412  20.0  3.97    0 0.6470 7.520  89.4  2.1398   5 264    13.0 388.37
## 263  0.52014  20.0  3.97    0 0.6470 8.398  91.5  2.2885   5 264    13.0 386.86
## 264  0.82526  20.0  3.97    0 0.6470 7.327  94.5  2.0788   5 264    13.0 393.42
## 265  0.55007  20.0  3.97    0 0.6470 7.206  91.6  1.9301   5 264    13.0 387.89
## 266  0.76162  20.0  3.97    0 0.6470 5.560  62.8  1.9865   5 264    13.0 392.40
## 267  0.78570  20.0  3.97    0 0.6470 7.014  84.6  2.1329   5 264    13.0 384.07
## 268  0.57834  20.0  3.97    0 0.5750 8.297  67.0  2.4216   5 264    13.0 384.54
## 269  0.54050  20.0  3.97    0 0.5750 7.470  52.6  2.8720   5 264    13.0 390.30
## 270  0.09065  20.0  6.96    1 0.4640 5.920  61.5  3.9175   3 223    18.6 391.34
## 271  0.29916  20.0  6.96    0 0.4640 5.856  42.1  4.4290   3 223    18.6 388.65
## 272  0.16211  20.0  6.96    0 0.4640 6.240  16.3  4.4290   3 223    18.6 396.90
## 273  0.11460  20.0  6.96    0 0.4640 6.538  58.7  3.9175   3 223    18.6 394.96
## 274  0.22188  20.0  6.96    1 0.4640 7.691  51.8  4.3665   3 223    18.6 390.77
## 275  0.05644  40.0  6.41    1 0.4470 6.758  32.9  4.0776   4 254    17.6 396.90
## 276  0.09604  40.0  6.41    0 0.4470 6.854  42.8  4.2673   4 254    17.6 396.90
## 277  0.10469  40.0  6.41    1 0.4470 7.267  49.0  4.7872   4 254    17.6 389.25
## 278  0.06127  40.0  6.41    1 0.4470 6.826  27.6  4.8628   4 254    17.6 393.45
## 279  0.07978  40.0  6.41    0 0.4470 6.482  32.1  4.1403   4 254    17.6 396.90
## 280  0.21038  20.0  3.33    0 0.4429 6.812  32.2  4.1007   5 216    14.9 396.90
## 281  0.03578  20.0  3.33    0 0.4429 7.820  64.5  4.6947   5 216    14.9 387.31
## 282  0.03705  20.0  3.33    0 0.4429 6.968  37.2  5.2447   5 216    14.9 392.23
## 283  0.06129  20.0  3.33    1 0.4429 7.645  49.7  5.2119   5 216    14.9 377.07
## 284  0.01501  90.0  1.21    1 0.4010 7.923  24.8  5.8850   1 198    13.6 395.52
## 285  0.00906  90.0  2.97    0 0.4000 7.088  20.8  7.3073   1 285    15.3 394.72
## 286  0.01096  55.0  2.25    0 0.3890 6.453  31.9  7.3073   1 300    15.3 394.72
## 287  0.01965  80.0  1.76    0 0.3850 6.230  31.5  9.0892   1 241    18.2 341.60
## 288  0.03871  52.5  5.32    0 0.4050 6.209  31.3  7.3172   6 293    16.6 396.90
## 289  0.04590  52.5  5.32    0 0.4050 6.315  45.6  7.3172   6 293    16.6 396.90
## 290  0.04297  52.5  5.32    0 0.4050 6.565  22.9  7.3172   6 293    16.6 371.72
## 291  0.03502  80.0  4.95    0 0.4110 6.861  27.9  5.1167   4 245    19.2 396.90
## 292  0.07886  80.0  4.95    0 0.4110 7.148  27.7  5.1167   4 245    19.2 396.90
## 293  0.03615  80.0  4.95    0 0.4110 6.630  23.4  5.1167   4 245    19.2 396.90
## 294  0.08265   0.0 13.92    0 0.4370 6.127  18.4  5.5027   4 289    16.0 396.90
## 295  0.08199   0.0 13.92    0 0.4370 6.009  42.3  5.5027   4 289    16.0 396.90
## 296  0.12932   0.0 13.92    0 0.4370 6.678  31.1  5.9604   4 289    16.0 396.90
## 297  0.05372   0.0 13.92    0 0.4370 6.549  51.0  5.9604   4 289    16.0 392.85
## 298  0.14103   0.0 13.92    0 0.4370 5.790  58.0  6.3200   4 289    16.0 396.90
## 299  0.06466  70.0  2.24    0 0.4000 6.345  20.1  7.8278   5 358    14.8 368.24
## 300  0.05561  70.0  2.24    0 0.4000 7.041  10.0  7.8278   5 358    14.8 371.58
## 301  0.04417  70.0  2.24    0 0.4000 6.871  47.4  7.8278   5 358    14.8 390.86
## 302  0.03537  34.0  6.09    0 0.4330 6.590  40.4  5.4917   7 329    16.1 395.75
```

```
## 303  0.09266  34.0  6.09     0 0.4330 6.495  18.4  5.4917   7 329     16.1 383.61
## 304  0.10000  34.0  6.09     0 0.4330 6.982  17.7  5.4917   7 329     16.1 390.43
## 305  0.05515  33.0  2.18     0 0.4720 7.236  41.1  4.0220   7 222     18.4 393.68
## 306  0.05479  33.0  2.18     0 0.4720 6.616  58.1  3.3700   7 222     18.4 393.36
## 307  0.07503  33.0  2.18     0 0.4720 7.420  71.9  3.0992   7 222     18.4 396.90
## 308  0.04932  33.0  2.18     0 0.4720 6.849  70.3  3.1827   7 222     18.4 396.90
## 309  0.49298   0.0  9.90     0 0.5440 6.635  82.5  3.3175   4 304     18.4 396.90
## 310  0.34940   0.0  9.90     0 0.5440 5.972  76.7  3.1025   4 304     18.4 396.24
## 311  2.63548   0.0  9.90     0 0.5440 4.973  37.8  2.5194   4 304     18.4 350.45
## 312  0.79041   0.0  9.90     0 0.5440 6.122  52.8  2.6403   4 304     18.4 396.90
## 313  0.26169   0.0  9.90     0 0.5440 6.023  90.4  2.8340   4 304     18.4 396.30
## 314  0.26938   0.0  9.90     0 0.5440 6.266  82.8  3.2628   4 304     18.4 393.39
## 315  0.36920   0.0  9.90     0 0.5440 6.567  87.3  3.6023   4 304     18.4 395.69
## 316  0.25356   0.0  9.90     0 0.5440 5.705  77.7  3.9450   4 304     18.4 396.42
## 317  0.31827   0.0  9.90     0 0.5440 5.914  83.2  3.9986   4 304     18.4 390.70
## 318  0.24522   0.0  9.90     0 0.5440 5.782  71.7  4.0317   4 304     18.4 396.90
## 319  0.40202   0.0  9.90     0 0.5440 6.382  67.2  3.5325   4 304     18.4 395.21
## 320  0.47547   0.0  9.90     0 0.5440 6.113  58.8  4.0019   4 304     18.4 396.23
## 321  0.16760   0.0  7.38     0 0.4930 6.426  52.3  4.5404   5 287     19.6 396.90
## 322  0.18159   0.0  7.38     0 0.4930 6.376  54.3  4.5404   5 287     19.6 396.90
## 323  0.35114   0.0  7.38     0 0.4930 6.041  49.9  4.7211   5 287     19.6 396.90
## 324  0.28392   0.0  7.38     0 0.4930 5.708  74.3  4.7211   5 287     19.6 391.13
## 325  0.34109   0.0  7.38     0 0.4930 6.415  40.1  4.7211   5 287     19.6 396.90
## 326  0.19186   0.0  7.38     0 0.4930 6.431  14.7  5.4159   5 287     19.6 393.68
## 327  0.30347   0.0  7.38     0 0.4930 6.312  28.9  5.4159   5 287     19.6 396.90
## 328  0.24103   0.0  7.38     0 0.4930 6.083  43.7  5.4159   5 287     19.6 396.90
## 329  0.06617   0.0  3.24     0 0.4600 5.868  25.8  5.2146   4 430     16.9 382.44
## 330  0.06724   0.0  3.24     0 0.4600 6.333  17.2  5.2146   4 430     16.9 375.21
## 331  0.04544   0.0  3.24     0 0.4600 6.144  32.2  5.8736   4 430     16.9 368.57
## 332  0.05023  35.0  6.06     0 0.4379 5.706  28.4  6.6407   1 304     16.9 394.02
## 333  0.03466  35.0  6.06     0 0.4379 6.031  23.3  6.6407   1 304     16.9 362.25
## 334  0.05083   0.0  5.19     0 0.5150 6.316  38.1  6.4584   5 224     20.2 389.71
## 335  0.03738   0.0  5.19     0 0.5150 6.310  38.5  6.4584   5 224     20.2 389.40
## 336  0.03961   0.0  5.19     0 0.5150 6.037  34.5  5.9853   5 224     20.2 396.90
## 337  0.03427   0.0  5.19     0 0.5150 5.869  46.3  5.2311   5 224     20.2 396.90
## 338  0.03041   0.0  5.19     0 0.5150 5.895  59.6  5.6150   5 224     20.2 394.81
## 339  0.03306   0.0  5.19     0 0.5150 6.059  37.3  4.8122   5 224     20.2 396.14
## 340  0.05497   0.0  5.19     0 0.5150 5.985  45.4  4.8122   5 224     20.2 396.90
## 341  0.06151   0.0  5.19     0 0.5150 5.968  58.5  4.8122   5 224     20.2 396.90
## 342  0.01301  35.0  1.52     0 0.4420 7.241  49.3  7.0379   1 284     15.5 394.74
## 343  0.02498   0.0  1.89     0 0.5180 6.540  59.7  6.2669   1 422     15.9 389.96
## 344  0.02543  55.0  3.78     0 0.4840 6.696  56.4  5.7321   5 370     17.6 396.90
## 345  0.03049  55.0  3.78     0 0.4840 6.874  28.1  6.4654   5 370     17.6 387.97
## 346  0.03113   0.0  4.39     0 0.4420 6.014  48.5  8.0136   3 352     18.8 385.64
## 347  0.06162   0.0  4.39     0 0.4420 5.898  52.3  8.0136   3 352     18.8 364.61
## 348  0.01870  85.0  4.15     0 0.4290 6.516  27.7  8.5353   4 351     17.9 392.43
## 349  0.01501  80.0  2.01     0 0.4350 6.635  29.7  8.3440   4 280     17.0 390.94
## 350  0.02899  40.0  1.25     0 0.4290 6.939  34.5  8.7921   1 335     19.7 389.85
## 351  0.06211  40.0  1.25     0 0.4290 6.490  44.4  8.7921   1 335     19.7 396.90
## 352  0.07950  60.0  1.69     0 0.4110 6.579  35.9 10.7103   4 411     18.3 370.78
## 353  0.07244  60.0  1.69     0 0.4110 5.884  18.5 10.7103   4 411     18.3 392.33
## 354  0.01709  90.0  2.02     0 0.4100 6.728  36.1 12.1265   5 187     17.0 384.46
## 355  0.04301  80.0  1.91     0 0.4130 5.663  21.9 10.5857   4 334     22.0 382.80
## 356  0.10659  80.0  1.91     0 0.4130 5.936  19.5 10.5857   4 334     22.0 376.04
```

```
## 357  8.98296   0.0 18.10    1 0.7700 6.212  97.4  2.1222  24 666    20.2 377.73
## 358  3.84970   0.0 18.10    1 0.7700 6.395  91.0  2.5052  24 666    20.2 391.34
## 359  5.20177   0.0 18.10    1 0.7700 6.127  83.4  2.7227  24 666    20.2 395.43
## 360  4.26131   0.0 18.10    0 0.7700 6.112  81.3  2.5091  24 666    20.2 390.74
## 361  4.54192   0.0 18.10    0 0.7700 6.398  88.0  2.5182  24 666    20.2 374.56
## 362  3.83684   0.0 18.10    0 0.7700 6.251  91.1  2.2955  24 666    20.2 350.65
## 363  3.67822   0.0 18.10    0 0.7700 5.362  96.2  2.1036  24 666    20.2 380.79
## 364  4.22239   0.0 18.10    1 0.7700 5.803  89.0  1.9047  24 666    20.2 353.04
## 365  3.47428   0.0 18.10    1 0.7180 8.780  82.9  1.9047  24 666    20.2 354.55
## 366  4.55587   0.0 18.10    0 0.7180 3.561  87.9  1.6132  24 666    20.2 354.70
## 367  3.69695   0.0 18.10    0 0.7180 4.963  91.4  1.7523  24 666    20.2 316.03
## 368 13.52220   0.0 18.10    0 0.6310 3.863 100.0  1.5106  24 666    20.2 131.42
## 369  4.89822   0.0 18.10    0 0.6310 4.970 100.0  1.3325  24 666    20.2 375.52
## 370  5.66998   0.0 18.10    1 0.6310 6.683  96.8  1.3567  24 666    20.2 375.33
## 371  6.53876   0.0 18.10    1 0.6310 7.016  97.5  1.2024  24 666    20.2 392.05
## 372  9.23230   0.0 18.10    0 0.6310 6.216 100.0  1.1691  24 666    20.2 366.15
## 373  8.26725   0.0 18.10    1 0.6680 5.875  89.6  1.1296  24 666    20.2 347.88
## 374 11.10810   0.0 18.10    0 0.6680 4.906 100.0  1.1742  24 666    20.2 396.90
## 375 18.49820   0.0 18.10    0 0.6680 4.138 100.0  1.1370  24 666    20.2 396.90
## 376 19.60910   0.0 18.10    0 0.6710 7.313  97.9  1.3163  24 666    20.2 396.90
## 377 15.28800   0.0 18.10    0 0.6710 6.649  93.3  1.3449  24 666    20.2 363.02
## 378  9.82349   0.0 18.10    0 0.6710 6.794  98.8  1.3580  24 666    20.2 396.90
## 379 23.64820   0.0 18.10    0 0.6710 6.380  96.2  1.3861  24 666    20.2 396.90
## 380 17.86670   0.0 18.10    0 0.6710 6.223 100.0  1.3861  24 666    20.2 393.74
## 381 88.97620   0.0 18.10    0 0.6710 6.968  91.9  1.4165  24 666    20.2 396.90
## 382 15.87440   0.0 18.10    0 0.6710 6.545  99.1  1.5192  24 666    20.2 396.90
## 383  9.18702   0.0 18.10    0 0.7000 5.536 100.0  1.5804  24 666    20.2 396.90
## 384  7.99248   0.0 18.10    0 0.7000 5.520 100.0  1.5331  24 666    20.2 396.90
## 385 20.08490   0.0 18.10    0 0.7000 4.368  91.2  1.4395  24 666    20.2 285.83
## 386 16.81180   0.0 18.10    0 0.7000 5.277  98.1  1.4261  24 666    20.2 396.90
## 387 24.39380   0.0 18.10    0 0.7000 4.652 100.0  1.4672  24 666    20.2 396.90
## 388 22.59710   0.0 18.10    0 0.7000 5.000  89.5  1.5184  24 666    20.2 396.90
## 389 14.33370   0.0 18.10    0 0.7000 4.880 100.0  1.5895  24 666    20.2 372.92
## 390  8.15174   0.0 18.10    0 0.7000 5.390  98.9  1.7281  24 666    20.2 396.90
## 391  6.96215   0.0 18.10    0 0.7000 5.713  97.0  1.9265  24 666    20.2 394.43
## 392  5.29305   0.0 18.10    0 0.7000 6.051  82.5  2.1678  24 666    20.2 378.38
## 393 11.57790   0.0 18.10    0 0.7000 5.036  97.0  1.7700  24 666    20.2 396.90
## 394  8.64476   0.0 18.10    0 0.6930 6.193  92.6  1.7912  24 666    20.2 396.90
## 395 13.35980   0.0 18.10    0 0.6930 5.887  94.7  1.7821  24 666    20.2 396.90
## 396  8.71675   0.0 18.10    0 0.6930 6.471  98.8  1.7257  24 666    20.2 391.98
## 397  5.87205   0.0 18.10    0 0.6930 6.405  96.0  1.6768  24 666    20.2 396.90
## 398  7.67202   0.0 18.10    0 0.6930 5.747  98.9  1.6334  24 666    20.2 393.10
## 399 38.35180   0.0 18.10    0 0.6930 5.453 100.0  1.4896  24 666    20.2 396.90
## 400  9.91655   0.0 18.10    0 0.6930 5.852  77.8  1.5004  24 666    20.2 338.16
## 401 25.04610   0.0 18.10    0 0.6930 5.987 100.0  1.5888  24 666    20.2 396.90
## 402 14.23620   0.0 18.10    0 0.6930 6.343 100.0  1.5741  24 666    20.2 396.90
## 403  9.59571   0.0 18.10    0 0.6930 6.404 100.0  1.6390  24 666    20.2 376.11
## 404 24.80170   0.0 18.10    0 0.6930 5.349  96.0  1.7028  24 666    20.2 396.90
## 405 41.52920   0.0 18.10    0 0.6930 5.531  85.4  1.6074  24 666    20.2 329.46
## 406 67.92080   0.0 18.10    0 0.6930 5.683 100.0  1.4254  24 666    20.2 384.97
## 407 20.71620   0.0 18.10    0 0.6590 4.138 100.0  1.1781  24 666    20.2 370.22
## 408 11.95110   0.0 18.10    0 0.6590 5.608 100.0  1.2852  24 666    20.2 332.09
## 409  7.40389   0.0 18.10    0 0.5970 5.617  97.9  1.4547  24 666    20.2 314.64
## 410 14.43830   0.0 18.10    0 0.5970 6.852 100.0  1.4655  24 666    20.2 179.36
```

```
## 411 51.13580   0.0 18.10    0 0.5970 5.757 100.0  1.4130  24 666    20.2   2.60
## 412 14.05070   0.0 18.10    0 0.5970 6.657 100.0  1.5275  24 666    20.2  35.05
## 413 18.81100   0.0 18.10    0 0.5970 4.628 100.0  1.5539  24 666    20.2  28.79
## 414 28.65580   0.0 18.10    0 0.5970 5.155 100.0  1.5894  24 666    20.2 210.97
## 415 45.74610   0.0 18.10    0 0.6930 4.519 100.0  1.6582  24 666    20.2  88.27
## 416 18.08460   0.0 18.10    0 0.6790 6.434 100.0  1.8347  24 666    20.2  27.25
## 417 10.83420   0.0 18.10    0 0.6790 6.782  90.8  1.8195  24 666    20.2  21.57
## 418 25.94060   0.0 18.10    0 0.6790 5.304  89.1  1.6475  24 666    20.2 127.36
## 419 73.53410   0.0 18.10    0 0.6790 5.957 100.0  1.8026  24 666    20.2  16.45
## 420 11.81230   0.0 18.10    0 0.7180 6.824  76.5  1.7940  24 666    20.2  48.45
## 421 11.08740   0.0 18.10    0 0.7180 6.411 100.0  1.8589  24 666    20.2 318.75
## 422  7.02259   0.0 18.10    0 0.7180 6.006  95.3  1.8746  24 666    20.2 319.98
## 423 12.04820   0.0 18.10    0 0.6140 5.648  87.6  1.9512  24 666    20.2 291.55
## 424  7.05042   0.0 18.10    0 0.6140 6.103  85.1  2.0218  24 666    20.2   2.52
## 425  8.79212   0.0 18.10    0 0.5840 5.565  70.6  2.0635  24 666    20.2   3.65
## 426 15.86030   0.0 18.10    0 0.6790 5.896  95.4  1.9096  24 666    20.2   7.68
## 427 12.24720   0.0 18.10    0 0.5840 5.837  59.7  1.9976  24 666    20.2  24.65
## 428 37.66190   0.0 18.10    0 0.6790 6.202  78.7  1.8629  24 666    20.2  18.82
## 429  7.36711   0.0 18.10    0 0.6790 6.193  78.1  1.9356  24 666    20.2  96.73
## 430  9.33889   0.0 18.10    0 0.6790 6.380  95.6  1.9682  24 666    20.2  60.72
## 431  8.49213   0.0 18.10    0 0.5840 6.348  86.1  2.0527  24 666    20.2  83.45
## 432 10.06230   0.0 18.10    0 0.5840 6.833  94.3  2.0882  24 666    20.2  81.33
## 433  6.44405   0.0 18.10    0 0.5840 6.425  74.8  2.2004  24 666    20.2  97.95
## 434  5.58107   0.0 18.10    0 0.7130 6.436  87.9  2.3158  24 666    20.2 100.19
## 435 13.91340   0.0 18.10    0 0.7130 6.208  95.0  2.2222  24 666    20.2 100.63
## 436 11.16040   0.0 18.10    0 0.7400 6.629  94.6  2.1247  24 666    20.2 109.85
## 437 14.42080   0.0 18.10    0 0.7400 6.461  93.3  2.0026  24 666    20.2  27.49
## 438 15.17720   0.0 18.10    0 0.7400 6.152 100.0  1.9142  24 666    20.2   9.32
## 439 13.67810   0.0 18.10    0 0.7400 5.935  87.9  1.8206  24 666    20.2  68.95
## 440  9.39063   0.0 18.10    0 0.7400 5.627  93.9  1.8172  24 666    20.2 396.90
## 441 22.05110   0.0 18.10    0 0.7400 5.818  92.4  1.8662  24 666    20.2 391.45
## 442  9.72418   0.0 18.10    0 0.7400 6.406  97.2  2.0651  24 666    20.2 385.96
## 443  5.66637   0.0 18.10    0 0.7400 6.219 100.0  2.0048  24 666    20.2 395.69
## 444  9.96654   0.0 18.10    0 0.7400 6.485 100.0  1.9784  24 666    20.2 386.73
## 445 12.80230   0.0 18.10    0 0.7400 5.854  96.6  1.8956  24 666    20.2 240.52
## 446 10.67180   0.0 18.10    0 0.7400 6.459  94.8  1.9879  24 666    20.2  43.06
## 447  6.28807   0.0 18.10    0 0.7400 6.341  96.4  2.0720  24 666    20.2 318.01
## 448  9.92485   0.0 18.10    0 0.7400 6.251  96.6  2.1980  24 666    20.2 388.52
## 449  9.32909   0.0 18.10    0 0.7130 6.185  98.7  2.2616  24 666    20.2 396.90
## 450  7.52601   0.0 18.10    0 0.7130 6.417  98.3  2.1850  24 666    20.2 304.21
## 451  6.71772   0.0 18.10    0 0.7130 6.749  92.6  2.3236  24 666    20.2   0.32
## 452  5.44114   0.0 18.10    0 0.7130 6.655  98.2  2.3552  24 666    20.2 355.29
## 453  5.09017   0.0 18.10    0 0.7130 6.297  91.8  2.3682  24 666    20.2 385.09
## 454  8.24809   0.0 18.10    0 0.7130 7.393  99.3  2.4527  24 666    20.2 375.87
## 455  9.51363   0.0 18.10    0 0.7130 6.728  94.1  2.4961  24 666    20.2   6.68
## 456  4.75237   0.0 18.10    0 0.7130 6.525  86.5  2.4358  24 666    20.2  50.92
## 457  4.66883   0.0 18.10    0 0.7130 5.976  87.9  2.5806  24 666    20.2  10.48
## 458  8.20058   0.0 18.10    0 0.7130 5.936  80.3  2.7792  24 666    20.2   3.50
## 459  7.75223   0.0 18.10    0 0.7130 6.301  83.7  2.7831  24 666    20.2 272.21
## 460  6.80117   0.0 18.10    0 0.7130 6.081  84.4  2.7175  24 666    20.2 396.90
## 461  4.81213   0.0 18.10    0 0.7130 6.701  90.0  2.5975  24 666    20.2 255.23
## 462  3.69311   0.0 18.10    0 0.7130 6.376  88.4  2.5671  24 666    20.2 391.43
## 463  6.65492   0.0 18.10    0 0.7130 6.317  83.0  2.7344  24 666    20.2 396.90
## 464  5.82115   0.0 18.10    0 0.7130 6.513  89.9  2.8016  24 666    20.2 393.82
```

```
## 465  7.83932   0.0 18.10    0 0.6550 6.209  65.4  2.9634  24 666    20.2 396.90
## 466  3.16360   0.0 18.10    0 0.6550 5.759  48.2  3.0665  24 666    20.2 334.40
## 467  3.77498   0.0 18.10    0 0.6550 5.952  84.7  2.8715  24 666    20.2  22.01
## 468  4.42228   0.0 18.10    0 0.5840 6.003  94.5  2.5403  24 666    20.2 331.29
## 469 15.57570   0.0 18.10    0 0.5800 5.926  71.0  2.9084  24 666    20.2 368.74
## 470 13.07510   0.0 18.10    0 0.5800 5.713  56.7  2.8237  24 666    20.2 396.90
## 471  4.34879   0.0 18.10    0 0.5800 6.167  84.0  3.0334  24 666    20.2 396.90
## 472  4.03841   0.0 18.10    0 0.5320 6.229  90.7  3.0993  24 666    20.2 395.33
## 473  3.56868   0.0 18.10    0 0.5800 6.437  75.0  2.8965  24 666    20.2 393.37
## 474  4.64689   0.0 18.10    0 0.6140 6.980  67.6  2.5329  24 666    20.2 374.68
## 475  8.05579   0.0 18.10    0 0.5840 5.427  95.4  2.4298  24 666    20.2 352.58
## 476  6.39312   0.0 18.10    0 0.5840 6.162  97.4  2.2060  24 666    20.2 302.76
## 477  4.87141   0.0 18.10    0 0.6140 6.484  93.6  2.3053  24 666    20.2 396.21
## 478 15.02340   0.0 18.10    0 0.6140 5.304  97.3  2.1007  24 666    20.2 349.48
## 479 10.23300   0.0 18.10    0 0.6140 6.185  96.7  2.1705  24 666    20.2 379.70
## 480 14.33370   0.0 18.10    0 0.6140 6.229  88.0  1.9512  24 666    20.2 383.32
## 481  5.82401   0.0 18.10    0 0.5320 6.242  64.7  3.4242  24 666    20.2 396.90
## 482  5.70818   0.0 18.10    0 0.5320 6.750  74.9  3.3317  24 666    20.2 393.07
## 483  5.73116   0.0 18.10    0 0.5320 7.061  77.0  3.4106  24 666    20.2 395.28
## 484  2.81838   0.0 18.10    0 0.5320 5.762  40.3  4.0983  24 666    20.2 392.92
## 485  2.37857   0.0 18.10    0 0.5830 5.871  41.9  3.7240  24 666    20.2 370.73
## 486  3.67367   0.0 18.10    0 0.5830 6.312  51.9  3.9917  24 666    20.2 388.62
## 487  5.69175   0.0 18.10    0 0.5830 6.114  79.8  3.5459  24 666    20.2 392.68
## 488  4.83567   0.0 18.10    0 0.5830 5.905  53.2  3.1523  24 666    20.2 388.22
## 489  0.15086   0.0 27.74    0 0.6090 5.454  92.7  1.8209   4 711    20.1 395.09
## 490  0.18337   0.0 27.74    0 0.6090 5.414  98.3  1.7554   4 711    20.1 344.05
## 491  0.20746   0.0 27.74    0 0.6090 5.093  98.0  1.8226   4 711    20.1 318.43
## 492  0.10574   0.0 27.74    0 0.6090 5.983  98.8  1.8681   4 711    20.1 390.11
## 493  0.11132   0.0 27.74    0 0.6090 5.983  83.5  2.1099   4 711    20.1 396.90
## 494  0.17331   0.0  9.69    0 0.5850 5.707  54.0  2.3817   6 391    19.2 396.90
## 495  0.27957   0.0  9.69    0 0.5850 5.926  42.6  2.3817   6 391    19.2 396.90
## 496  0.17899   0.0  9.69    0 0.5850 5.670  28.8  2.7986   6 391    19.2 393.29
## 497  0.28960   0.0  9.69    0 0.5850 5.390  72.9  2.7986   6 391    19.2 396.90
## 498  0.26838   0.0  9.69    0 0.5850 5.794  70.6  2.8927   6 391    19.2 396.90
## 499  0.23912   0.0  9.69    0 0.5850 6.019  65.3  2.4091   6 391    19.2 396.90
## 500  0.17783   0.0  9.69    0 0.5850 5.569  73.5  2.3999   6 391    19.2 395.77
## 501  0.22438   0.0  9.69    0 0.5850 6.027  79.7  2.4982   6 391    19.2 396.90
## 502  0.06263   0.0 11.93    0 0.5730 6.593  69.1  2.4786   1 273    21.0 391.99
## 503  0.04527   0.0 11.93    0 0.5730 6.120  76.7  2.2875   1 273    21.0 396.90
## 504  0.06076   0.0 11.93    0 0.5730 6.976  91.0  2.1675   1 273    21.0 396.90
## 505  0.10959   0.0 11.93    0 0.5730 6.794  89.3  2.3889   1 273    21.0 393.45
## 506  0.04741   0.0 11.93    0 0.5730 6.030  80.8  2.5050   1 273    21.0 396.90
##     lstat medv
## 1    4.98 24.0
## 2    9.14 21.6
## 3    4.03 34.7
## 4    2.94 33.4
## 5    5.33 36.2
## 6    5.21 28.7
## 7   12.43 22.9
## 8   19.15 27.1
## 9   29.93 16.5
## 10  17.10 18.9
## 11  20.45 15.0
```

```
## 12  13.27 18.9
## 13  15.71 21.7
## 14   8.26 20.4
## 15  10.26 18.2
## 16   8.47 19.9
## 17   6.58 23.1
## 18  14.67 17.5
## 19  11.69 20.2
## 20  11.28 18.2
## 21  21.02 13.6
## 22  13.83 19.6
## 23  18.72 15.2
## 24  19.88 14.5
## 25  16.30 15.6
## 26  16.51 13.9
## 27  14.81 16.6
## 28  17.28 14.8
## 29  12.80 18.4
## 30  11.98 21.0
## 31  22.60 12.7
## 32  13.04 14.5
## 33  27.71 13.2
## 34  18.35 13.1
## 35  20.34 13.5
## 36   9.68 18.9
## 37  11.41 20.0
## 38   8.77 21.0
## 39  10.13 24.7
## 40   4.32 30.8
## 41   1.98 34.9
## 42   4.84 26.6
## 43   5.81 25.3
## 44   7.44 24.7
## 45   9.55 21.2
## 46  10.21 19.3
## 47  14.15 20.0
## 48  18.80 16.6
## 49  30.81 14.4
## 50  16.20 19.4
## 51  13.45 19.7
## 52   9.43 20.5
## 53   5.28 25.0
## 54   8.43 23.4
## 55  14.80 18.9
## 56   4.81 35.4
## 57   5.77 24.7
## 58   3.95 31.6
## 59   6.86 23.3
## 60   9.22 19.6
## 61  13.15 18.7
## 62  14.44 16.0
## 63   6.73 22.2
## 64   9.50 25.0
## 65   8.05 33.0
```

```
## 66   4.67 23.5
## 67  10.24 19.4
## 68   8.10 22.0
## 69  13.09 17.4
## 70   8.79 20.9
## 71   6.72 24.2
## 72   9.88 21.7
## 73   5.52 22.8
## 74   7.54 23.4
## 75   6.78 24.1
## 76   8.94 21.4
## 77  11.97 20.0
## 78  10.27 20.8
## 79  12.34 21.2
## 80   9.10 20.3
## 81   5.29 28.0
## 82   7.22 23.9
## 83   6.72 24.8
## 84   7.51 22.9
## 85   9.62 23.9
## 86   6.53 26.6
## 87  12.86 22.5
## 88   8.44 22.2
## 89   5.50 23.6
## 90   5.70 28.7
## 91   8.81 22.6
## 92   8.20 22.0
## 93   8.16 22.9
## 94   6.21 25.0
## 95  10.59 20.6
## 96   6.65 28.4
## 97  11.34 21.4
## 98   4.21 38.7
## 99   3.57 43.8
## 100  6.19 33.2
## 101  9.42 27.5
## 102  7.67 26.5
## 103 10.63 18.6
## 104 13.44 19.3
## 105 12.33 20.1
## 106 16.47 19.5
## 107 18.66 19.5
## 108 14.09 20.4
## 109 12.27 19.8
## 110 15.55 19.4
## 111 13.00 21.7
## 112 10.16 22.8
## 113 16.21 18.8
## 114 17.09 18.7
## 115 10.45 18.5
## 116 15.76 18.3
## 117 12.04 21.2
## 118 10.30 19.2
## 119 15.37 20.4
```

```
## 120 13.61 19.3
## 121 14.37 22.0
## 122 14.27 20.3
## 123 17.93 20.5
## 124 25.41 17.3
## 125 17.58 18.8
## 126 14.81 21.4
## 127 27.26 15.7
## 128 17.19 16.2
## 129 15.39 18.0
## 130 18.34 14.3
## 131 12.60 19.2
## 132 12.26 19.6
## 133 11.12 23.0
## 134 15.03 18.4
## 135 17.31 15.6
## 136 16.96 18.1
## 137 16.90 17.4
## 138 14.59 17.1
## 139 21.32 13.3
## 140 18.46 17.8
## 141 24.16 14.0
## 142 34.41 14.4
## 143 26.82 13.4
## 144 26.42 15.6
## 145 29.29 11.8
## 146 27.80 13.8
## 147 16.65 15.6
## 148 29.53 14.6
## 149 28.32 17.8
## 150 21.45 15.4
## 151 14.10 21.5
## 152 13.28 19.6
## 153 12.12 15.3
## 154 15.79 19.4
## 155 15.12 17.0
## 156 15.02 15.6
## 157 16.14 13.1
## 158  4.59 41.3
## 159  6.43 24.3
## 160  7.39 23.3
## 161  5.50 27.0
## 162  1.73 50.0
## 163  1.92 50.0
## 164  3.32 50.0
## 165 11.64 22.7
## 166  9.81 25.0
## 167  3.70 50.0
## 168 12.14 23.8
## 169 11.10 23.8
## 170 11.32 22.3
## 171 14.43 17.4
## 172 12.03 19.1
## 173 14.69 23.1
```

```
## 174  9.04 23.6
## 175  9.64 22.6
## 176  5.33 29.4
## 177 10.11 23.2
## 178  6.29 24.6
## 179  6.92 29.9
## 180  5.04 37.2
## 181  7.56 39.8
## 182  9.45 36.2
## 183  4.82 37.9
## 184  5.68 32.5
## 185 13.98 26.4
## 186 13.15 29.6
## 187  4.45 50.0
## 188  6.68 32.0
## 189  4.56 29.8
## 190  5.39 34.9
## 191  5.10 37.0
## 192  4.69 30.5
## 193  2.87 36.4
## 194  5.03 31.1
## 195  4.38 29.1
## 196  2.97 50.0
## 197  4.08 33.3
## 198  8.61 30.3
## 199  6.62 34.6
## 200  4.56 34.9
## 201  4.45 32.9
## 202  7.43 24.1
## 203  3.11 42.3
## 204  3.81 48.5
## 205  2.88 50.0
## 206 10.87 22.6
## 207 10.97 24.4
## 208 18.06 22.5
## 209 14.66 24.4
## 210 23.09 20.0
## 211 17.27 21.7
## 212 23.98 19.3
## 213 16.03 22.4
## 214  9.38 28.1
## 215 29.55 23.7
## 216  9.47 25.0
## 217 13.51 23.3
## 218  9.69 28.7
## 219 17.92 21.5
## 220 10.50 23.0
## 221  9.71 26.7
## 222 21.46 21.7
## 223  9.93 27.5
## 224  7.60 30.1
## 225  4.14 44.8
## 226  4.63 50.0
## 227  3.13 37.6
```

```
## 228  6.36 31.6
## 229  3.92 46.7
## 230  3.76 31.5
## 231 11.65 24.3
## 232  5.25 31.7
## 233  2.47 41.7
## 234  3.95 48.3
## 235  8.05 29.0
## 236 10.88 24.0
## 237  9.54 25.1
## 238  4.73 31.5
## 239  6.36 23.7
## 240  7.37 23.3
## 241 11.38 22.0
## 242 12.40 20.1
## 243 11.22 22.2
## 244  5.19 23.7
## 245 12.50 17.6
## 246 18.46 18.5
## 247  9.16 24.3
## 248 10.15 20.5
## 249  9.52 24.5
## 250  6.56 26.2
## 251  5.90 24.4
## 252  3.59 24.8
## 253  3.53 29.6
## 254  3.54 42.8
## 255  6.57 21.9
## 256  9.25 20.9
## 257  3.11 44.0
## 258  5.12 50.0
## 259  7.79 36.0
## 260  6.90 30.1
## 261  9.59 33.8
## 262  7.26 43.1
## 263  5.91 48.8
## 264 11.25 31.0
## 265  8.10 36.5
## 266 10.45 22.8
## 267 14.79 30.7
## 268  7.44 50.0
## 269  3.16 43.5
## 270 13.65 20.7
## 271 13.00 21.1
## 272  6.59 25.2
## 273  7.73 24.4
## 274  6.58 35.2
## 275  3.53 32.4
## 276  2.98 32.0
## 277  6.05 33.2
## 278  4.16 33.1
## 279  7.19 29.1
## 280  4.85 35.1
## 281  3.76 45.4
```

```
## 282  4.59 35.4
## 283  3.01 46.0
## 284  3.16 50.0
## 285  7.85 32.2
## 286  8.23 22.0
## 287 12.93 20.1
## 288  7.14 23.2
## 289  7.60 22.3
## 290  9.51 24.8
## 291  3.33 28.5
## 292  3.56 37.3
## 293  4.70 27.9
## 294  8.58 23.9
## 295 10.40 21.7
## 296  6.27 28.6
## 297  7.39 27.1
## 298 15.84 20.3
## 299  4.97 22.5
## 300  4.74 29.0
## 301  6.07 24.8
## 302  9.50 22.0
## 303  8.67 26.4
## 304  4.86 33.1
## 305  6.93 36.1
## 306  8.93 28.4
## 307  6.47 33.4
## 308  7.53 28.2
## 309  4.54 22.8
## 310  9.97 20.3
## 311 12.64 16.1
## 312  5.98 22.1
## 313 11.72 19.4
## 314  7.90 21.6
## 315  9.28 23.8
## 316 11.50 16.2
## 317 18.33 17.8
## 318 15.94 19.8
## 319 10.36 23.1
## 320 12.73 21.0
## 321  7.20 23.8
## 322  6.87 23.1
## 323  7.70 20.4
## 324 11.74 18.5
## 325  6.12 25.0
## 326  5.08 24.6
## 327  6.15 23.0
## 328 12.79 22.2
## 329  9.97 19.3
## 330  7.34 22.6
## 331  9.09 19.8
## 332 12.43 17.1
## 333  7.83 19.4
## 334  5.68 22.2
## 335  6.75 20.7
```

```
## 336  8.01 21.1
## 337  9.80 19.5
## 338 10.56 18.5
## 339  8.51 20.6
## 340  9.74 19.0
## 341  9.29 18.7
## 342  5.49 32.7
## 343  8.65 16.5
## 344  7.18 23.9
## 345  4.61 31.2
## 346 10.53 17.5
## 347 12.67 17.2
## 348  6.36 23.1
## 349  5.99 24.5
## 350  5.89 26.6
## 351  5.98 22.9
## 352  5.49 24.1
## 353  7.79 18.6
## 354  4.50 30.1
## 355  8.05 18.2
## 356  5.57 20.6
## 357 17.60 17.8
## 358 13.27 21.7
## 359 11.48 22.7
## 360 12.67 22.6
## 361  7.79 25.0
## 362 14.19 19.9
## 363 10.19 20.8
## 364 14.64 16.8
## 365  5.29 21.9
## 366  7.12 27.5
## 367 14.00 21.9
## 368 13.33 23.1
## 369  3.26 50.0
## 370  3.73 50.0
## 371  2.96 50.0
## 372  9.53 50.0
## 373  8.88 50.0
## 374 34.77 13.8
## 375 37.97 13.8
## 376 13.44 15.0
## 377 23.24 13.9
## 378 21.24 13.3
## 379 23.69 13.1
## 380 21.78 10.2
## 381 17.21 10.4
## 382 21.08 10.9
## 383 23.60 11.3
## 384 24.56 12.3
## 385 30.63  8.8
## 386 30.81  7.2
## 387 28.28 10.5
## 388 31.99  7.4
## 389 30.62 10.2
```

```
## 390 20.85 11.5
## 391 17.11 15.1
## 392 18.76 23.2
## 393 25.68  9.7
## 394 15.17 13.8
## 395 16.35 12.7
## 396 17.12 13.1
## 397 19.37 12.5
## 398 19.92  8.5
## 399 30.59  5.0
## 400 29.97  6.3
## 401 26.77  5.6
## 402 20.32  7.2
## 403 20.31 12.1
## 404 19.77  8.3
## 405 27.38  8.5
## 406 22.98  5.0
## 407 23.34 11.9
## 408 12.13 27.9
## 409 26.40 17.2
## 410 19.78 27.5
## 411 10.11 15.0
## 412 21.22 17.2
## 413 34.37 17.9
## 414 20.08 16.3
## 415 36.98  7.0
## 416 29.05  7.2
## 417 25.79  7.5
## 418 26.64 10.4
## 419 20.62  8.8
## 420 22.74  8.4
## 421 15.02 16.7
## 422 15.70 14.2
## 423 14.10 20.8
## 424 23.29 13.4
## 425 17.16 11.7
## 426 24.39  8.3
## 427 15.69 10.2
## 428 14.52 10.9
## 429 21.52 11.0
## 430 24.08  9.5
## 431 17.64 14.5
## 432 19.69 14.1
## 433 12.03 16.1
## 434 16.22 14.3
## 435 15.17 11.7
## 436 23.27 13.4
## 437 18.05  9.6
## 438 26.45  8.7
## 439 34.02  8.4
## 440 22.88 12.8
## 441 22.11 10.5
## 442 19.52 17.1
## 443 16.59 18.4
```

```
## 444 18.85 15.4
## 445 23.79 10.8
## 446 23.98 11.8
## 447 17.79 14.9
## 448 16.44 12.6
## 449 18.13 14.1
## 450 19.31 13.0
## 451 17.44 13.4
## 452 17.73 15.2
## 453 17.27 16.1
## 454 16.74 17.8
## 455 18.71 14.9
## 456 18.13 14.1
## 457 19.01 12.7
## 458 16.94 13.5
## 459 16.23 14.9
## 460 14.70 20.0
## 461 16.42 16.4
## 462 14.65 17.7
## 463 13.99 19.5
## 464 10.29 20.2
## 465 13.22 21.4
## 466 14.13 19.9
## 467 17.15 19.0
## 468 21.32 19.1
## 469 18.13 19.1
## 470 14.76 20.1
## 471 16.29 19.9
## 472 12.87 19.6
## 473 14.36 23.2
## 474 11.66 29.8
## 475 18.14 13.8
## 476 24.10 13.3
## 477 18.68 16.7
## 478 24.91 12.0
## 479 18.03 14.6
## 480 13.11 21.4
## 481 10.74 23.0
## 482  7.74 23.7
## 483  7.01 25.0
## 484 10.42 21.8
## 485 13.34 20.6
## 486 10.58 21.2
## 487 14.98 19.1
## 488 11.45 20.6
## 489 18.06 15.2
## 490 23.97  7.0
## 491 29.68  8.1
## 492 18.07 13.6
## 493 13.35 20.1
## 494 12.01 21.8
## 495 13.59 24.5
## 496 17.60 23.1
## 497 21.14 19.7
```

```
## 498 14.10 18.3
## 499 12.92 21.2
## 500 15.10 17.5
## 501 14.33 16.8
## 502  9.67 22.4
## 503  9.08 20.6
## 504  5.64 23.9
## 505  6.48 22.0
## 506  7.88 11.9
```

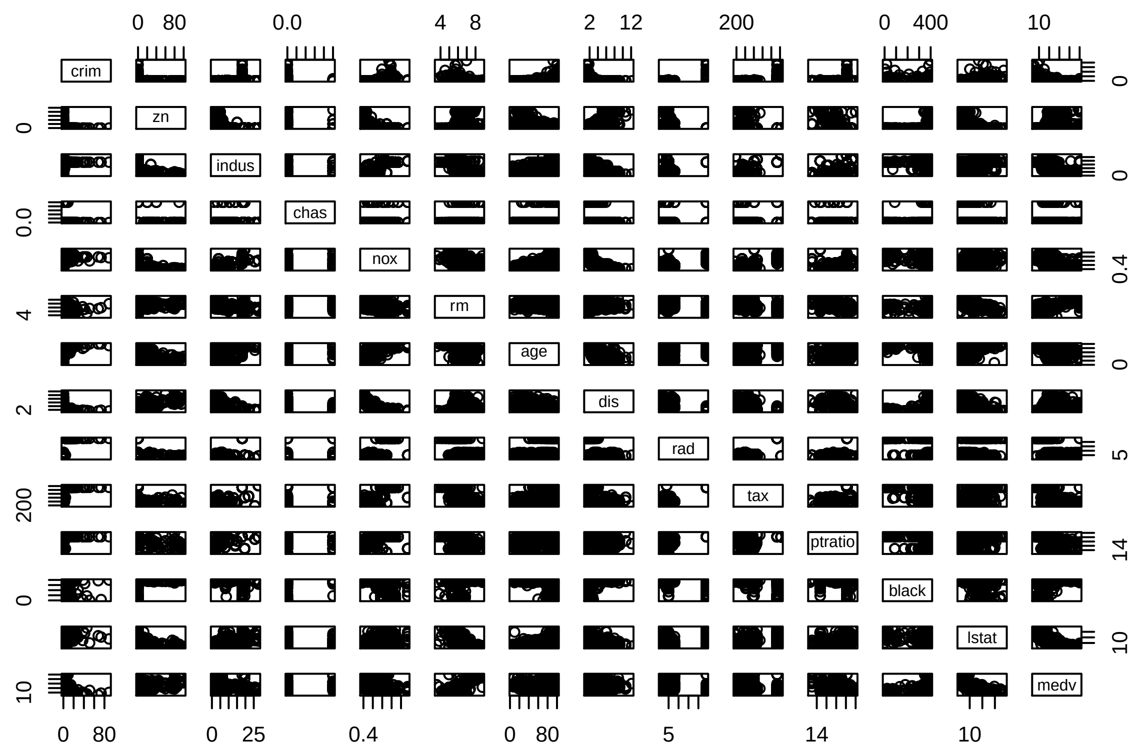
```r
?Boston
```

```
## starting httpd help server ... done
```

There are 506 rows and 14 columns in the Boston data set. The columns represent various attributes about housing in Boston, ranging from crime rate to all the way to the actual values of those homes. The rows represent suburbs in Boston and the corresponding values of the 14 features.

Chapter 2 - 10b

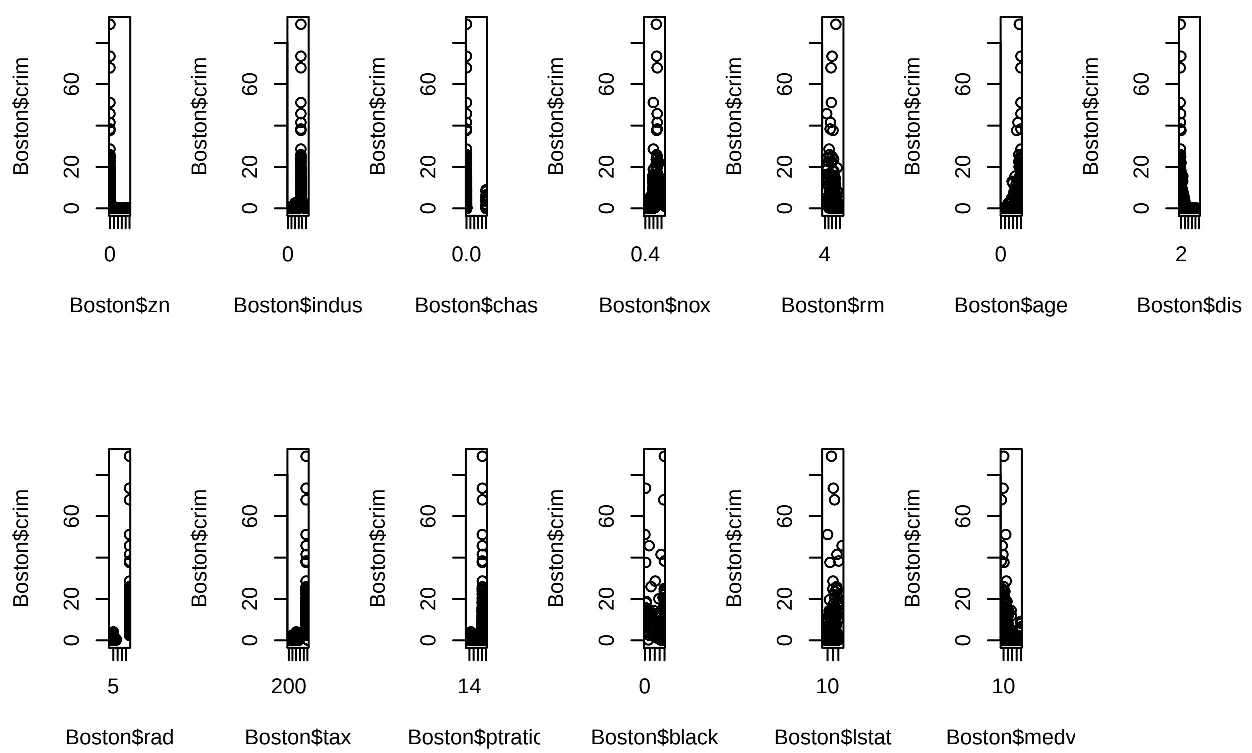
```r
pairs(Boston)
```



The scatter plots display the relationships of the 14 variables with respect to each other. Many of the factors with respect to each other do not seem to have any discernible relationship.However there is a negative relationship with medv and lstat and a rather negative logarithmic relationship between dis and nox. One positive relationship looks to be between medv and rm.

Chapter 2 - 10c

```r
par(mfrow = c(2,7))

plot(Boston$zn, Boston$crim)
plot(Boston$indus, Boston$crim)
plot(Boston$chas, Boston$crim)
plot(Boston$nox, Boston$crim)
plot(Boston$rm, Boston$crim)
plot(Boston$age, Boston$crim)
plot(Boston$dis, Boston$crim)
plot(Boston$rad, Boston$crim)
plot(Boston$tax, Boston$crim)
plot(Boston$ptratio, Boston$crim)
plot(Boston$black, Boston$crim)
plot(Boston$lstat, Boston$crim)
plot(Boston$medv, Boston$crim)
```



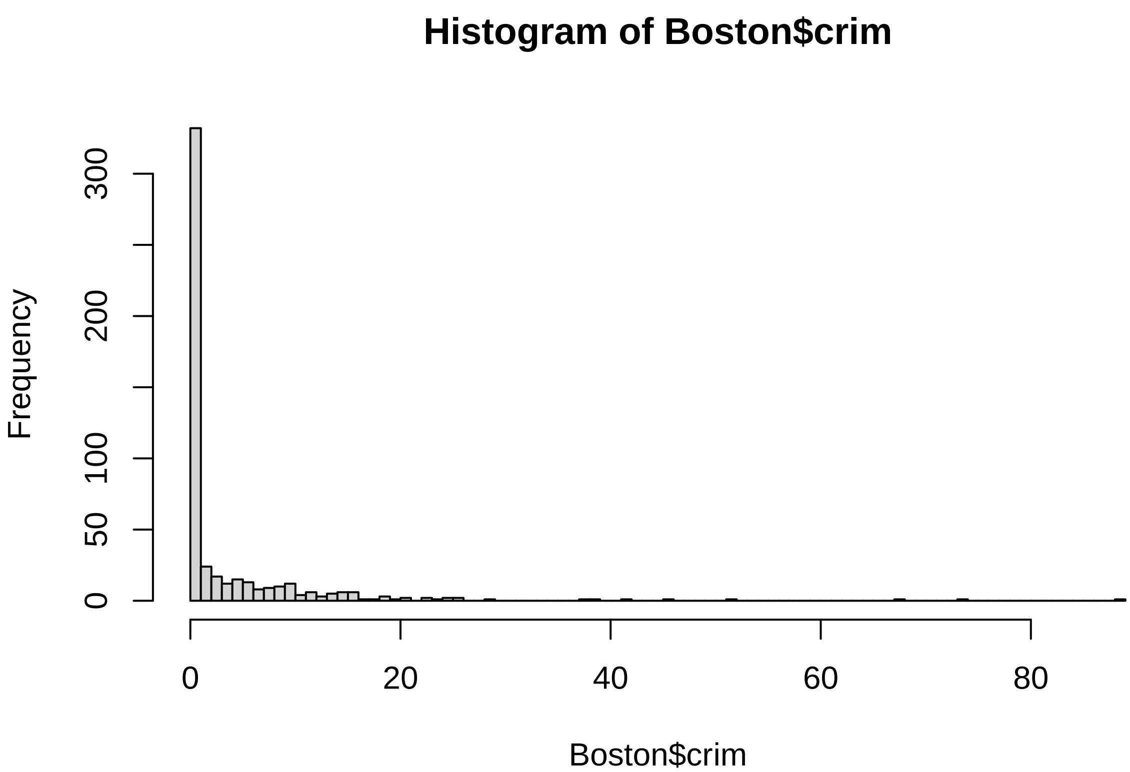
It seems that close by to the employment centres there is a high crime rate, while farther away, the crime capita rate goes down. The lower median values of houses also have higher crime rates than those houses with higher median values. With higher proportion of owners living in units built prior to 1940, the crime rate also seems to increase. Areas with large concentration of residential lots greater than 25,000 sq ft have lower crime rates. Where the tract bounds the Charles River, crime rate is low, otherwise it seems to be higher. A pupil-teacher ratio of approximately 20% shows a crime rate that is significantly larger.

Chapter 2 - 10d

```r
summary(Boston$crim)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  0.00632  0.08204  0.25651  3.61352  3.67708 88.97620
```

```r
par(mfrow = c(1,1))

hist(Boston$crim, 100)
```



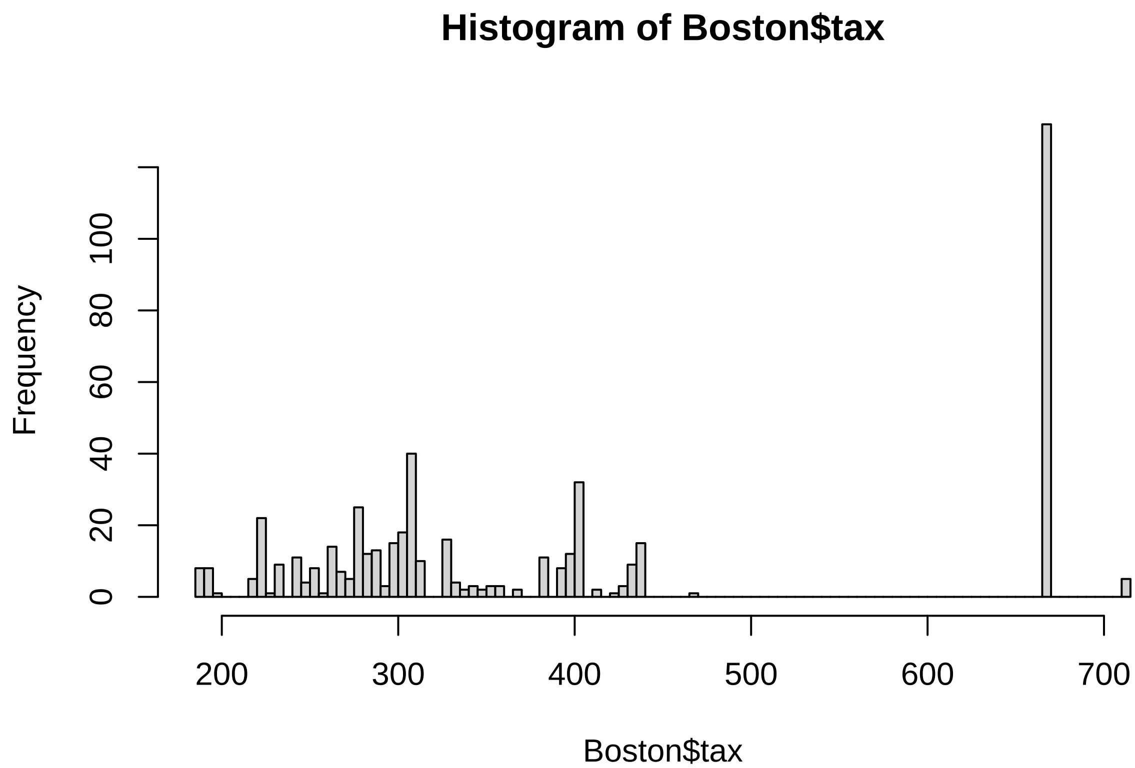

```r
range(Boston$crim)
```

```
## [1]  0.00632 88.97620
```

```r
nrow(Boston[Boston$crim > 20,])
```

```
## [1] 18
```

```r
hist(Boston$tax, 100)
```



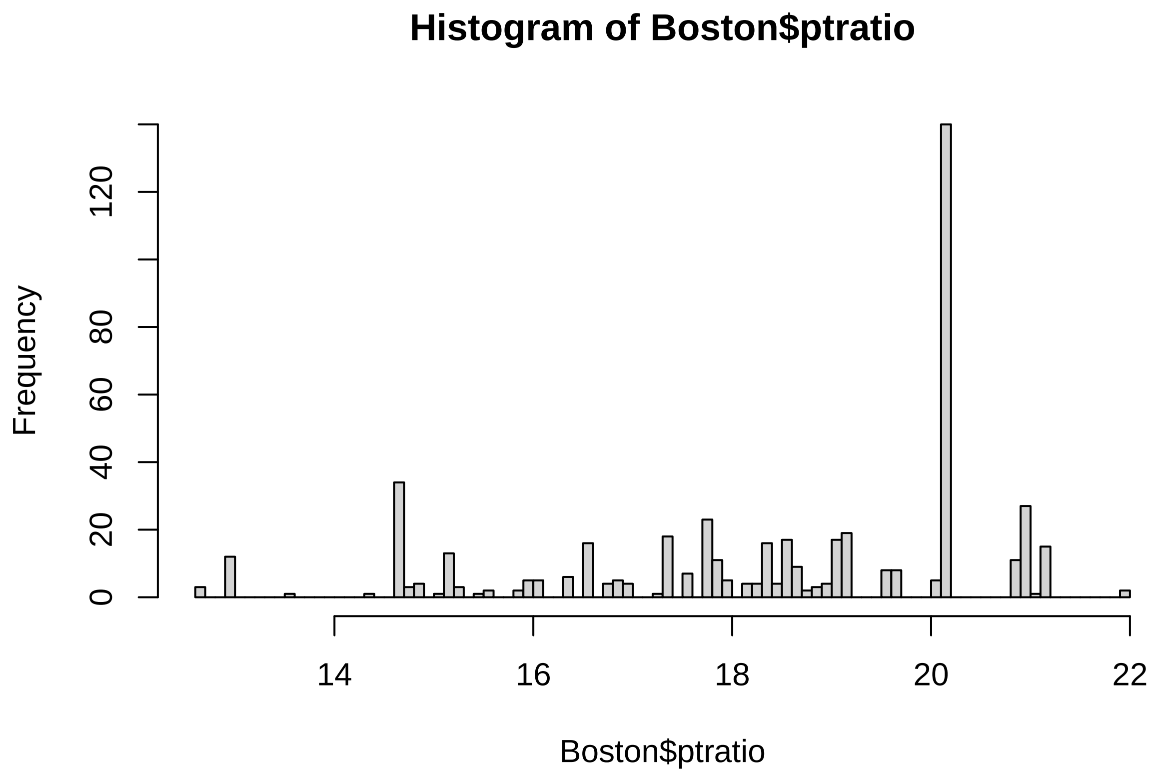

```
range(Boston$tax)
```

```
## [1] 187 711
```

```
nrow(Boston[Boston$tax > 400,])
```

```
## [1] 200
```

```
hist(Boston$ptratio, 100)
```



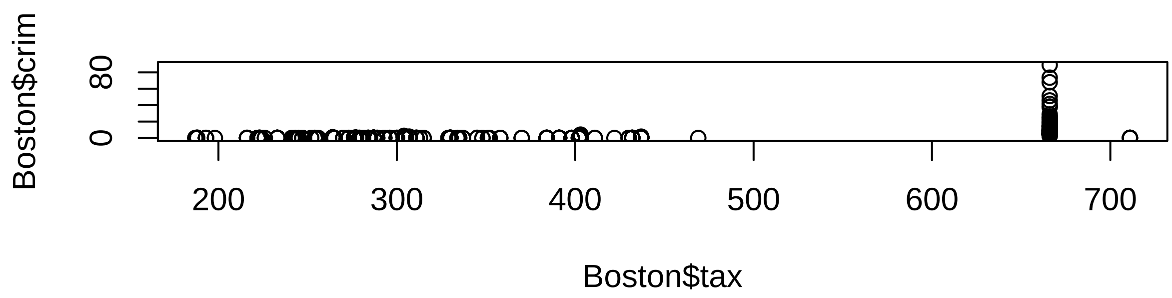

```r
range(Boston$ptratio)
```

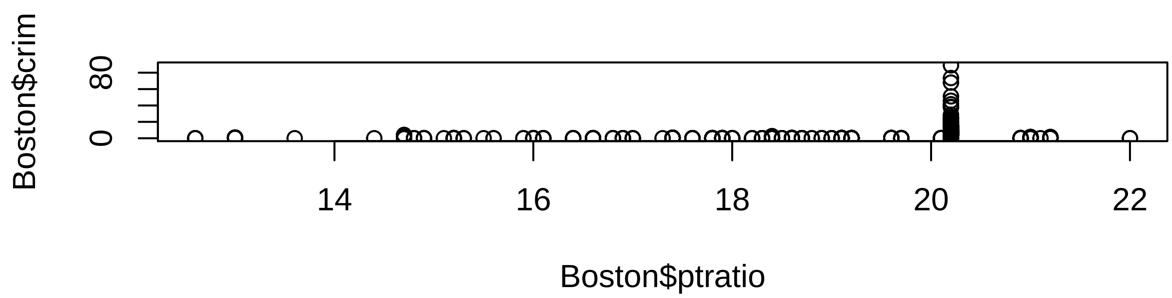
```
## [1] 12.6 22.0
```

```r
nrow(Boston[Boston$ptratio == 20.2,])
```

```
## [1] 140
```

```r
par(mfrow = c(2,1))

plot(Boston$tax, Boston$crim)
plot(Boston$ptratio, Boston$crim)
```





Certain area with a specific tax rate of approximately 670 seem to have a much higher crime rate than the rest of town.This could also be because, according to the histogram, the most frequency of suburbs have a tax rate of approximately 670. The crime rate for a ptratio of 20.2 is also disproportionately higher than the rest of town, but that might also be because a ptratio of 20.2 also corresponds to the highest frequency in Boston. The frequency of "high" crime rate, that is greater than 20, is relatively small being only 18. A reason why large crime rates show up for ptratio of 20.2 and a tax rate for 670 might be because those frequencies are much higher than the rest, and would show more variation.

Chapter 2 - 10e

```r
summary(Boston$chas)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.00000 0.00000 0.06917 0.00000 1.00000
```

```r
nrow(Boston[Boston$chas == 0,])
```

```
## [1] 471
```

```r
nrow(Boston[Boston$chas == 1,])
```

```
## [1] 35
```

Only 35 suburbs in this data set bound the Charles River.

Chapter 2 - 10f

```r
summary(Boston$ptratio)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   12.60   17.40   19.05   18.46   20.20   22.00
```

The median pupil-teacher ratio among the data set is 19.05.

Chapter 2 - 10g

```r
summary(Boston)
```

```
##       crim                zn             indus            chas        
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000  
##  1st Qu.: 0.08205   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000  
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000  
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917  
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000  
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000  
##       nox               rm             age              dis        
##  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130  
##  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100  
##  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207  
##  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795  
##  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188  
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127  
##       rad              tax           ptratio          black       
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32  
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38  
##  Median : 5.000   Median :330.0   Median :19.05   Median :391.44  
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67  
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23  
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90  
##      lstat            medv      
##  Min.   : 1.73   Min.   : 5.00  
##  1st Qu.: 6.95   1st Qu.:17.02  
##  Median :11.36   Median :21.20  
##  Mean   :12.65   Mean   :22.53  
##  3rd Qu.:16.95   3rd Qu.:25.00  
##  Max.   :37.97   Max.   :50.00
```

```r
lowest_median_value <- Boston[Boston$medv == min(Boston$medv),]

t(lowest_median_value)
```

```
##              399      406
## crim     38.3518  67.9208
## zn        0.0000   0.0000
## indus    18.1000  18.1000
## chas      0.0000   0.0000
## nox       0.6930   0.6930
## rm        5.4530   5.6830
## age     100.0000 100.0000
## dis       1.4896   1.4254
## rad      24.0000  24.0000
## tax     666.0000 666.0000
## ptratio  20.2000  20.2000
## black   396.9000 384.9700
```

```
## lstat    30.5900   22.9800
## medv      5.0000    5.0000
```

```r
range(Boston$crim)
```

```
## [1]  0.00632 88.97620
```

```r
median(Boston$crim)
```

```
## [1] 0.25651
```

```r
range(Boston$age)
```

```
## [1]   2.9 100.0
```

```r
median(Boston$age)
```

```
## [1] 77.5
```

```r
range(Boston$ptratio)
```

```
## [1] 12.6 22.0
```

```r
median(Boston$ptratio)
```

```
## [1] 19.05
```

```r
range(Boston$lstat)
```

```
## [1]  1.73 37.97
```

```r
median(Boston$lstat)
```

```
## [1] 11.36
```

There are two suburbs with the lowest median value of owner-occupied homes. The median value of these homes is $5,000.The corresponding crime rates are 38.3518 and 67.9208. The proportion of units that are built before 1940 for both suburbs is 100%. The pupil-teacher ratio for both suburbs is 20.2. The lower status of the population for the first suburb is 30.59%, and for the other, it is 22.98%. Suburbs with low median house values will tend to have higher crime rates, older houses, and higher percentage of the lower status of the population.

Chapter 2 - 10h

```r
summary(Boston$rm)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.561   5.886   6.208   6.285   6.623   8.780
```

```r
nrow(Boston[Boston$rm > 7,])
```

```
## [1] 64
```

```r
nrow(Boston[Boston$rm > 8,])
```

```
## [1] 13
```

```r
dwellings_more_than_eight <- Boston[Boston$rm > 8,]

dwellings_more_than_eight
```

```
##        crim zn indus chas    nox    rm  age    dis rad tax ptratio  black lstat
## 98  0.12083  0  2.89    0 0.4450 8.069 76.0 3.4952   2 276    18.0 396.90  4.21
```

```
## 164 1.51902  0 19.58    1 0.6050 8.375 93.9 2.1620   5 403    14.7 388.45  3.32
## 205 0.02009 95  2.68    0 0.4161 8.034 31.9 5.1180   4 224    14.7 390.55  2.88
## 225 0.31533  0  6.20    0 0.5040 8.266 78.3 2.8944   8 307    17.4 385.05  4.14
## 226 0.52693  0  6.20    0 0.5040 8.725 83.0 2.8944   8 307    17.4 382.00  4.63
## 227 0.38214  0  6.20    0 0.5040 8.040 86.5 3.2157   8 307    17.4 387.38  3.13
## 233 0.57529  0  6.20    0 0.5070 8.337 73.3 3.8384   8 307    17.4 385.91  2.47
## 234 0.33147  0  6.20    0 0.5070 8.247 70.4 3.6519   8 307    17.4 378.95  3.95
## 254 0.36894 22  5.86    0 0.4310 8.259  8.4 8.9067   7 330    19.1 396.90  3.54
## 258 0.61154 20  3.97    0 0.6470 8.704 86.9 1.8010   5 264    13.0 389.70  5.12
## 263 0.52014 20  3.97    0 0.6470 8.398 91.5 2.2885   5 264    13.0 386.86  5.91
## 268 0.57834 20  3.97    0 0.5750 8.297 67.0 2.4216   5 264    13.0 384.54  7.44
## 365 3.47428  0 18.10    1 0.7180 8.780 82.9 1.9047  24 666    20.2 354.55  5.29
##     medv
## 98  38.7
## 164 50.0
## 205 50.0
## 225 44.8
## 226 50.0
## 227 37.6
## 233 41.7
## 234 48.3
## 254 42.8
## 258 50.0
## 263 48.8
## 268 50.0
## 365 21.9
```

```r
median(dwellings_more_than_eight$crim)
```

```
## [1] 0.52014
```

```r
median(dwellings_more_than_eight$medv)
```

```
## [1] 48.3
```

```r
median(dwellings_more_than_eight$lstat)
```

```
## [1] 4.14
```

```r
median(dwellings_more_than_eight$age)
```

```
## [1] 78.3
```

There are 64 suburbs averaging more than 7 rooms per dwelling and 13 suburbs averaging more than 8 rooms per dwelling. The median crime rate is 0.7187954, the median home value is $48,300, the median lower status of the population is 4.14% and the median proportion of houses older than 1940 is 78.3%. It seems that suburbs that average more than 8 rooms per dwelling tend to have higher home values, lower crime rates, and lower proportion of lower status of the population.

Chapter 3 - 15

Chapter 3 - 15a

```r
rm(list = ls())

library(MASS)
```

```r
names(Boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

```r
crim.zn = lm(crim~zn, data = Boston)
crim.indus = lm(crim~indus, data = Boston)
crim.chas = lm(crim~chas, data = Boston)
crim.nox = lm(crim~nox, data = Boston)
crim.rm = lm(crim~rm, data = Boston)
crim.age = lm(crim~age, data = Boston)
crim.dis = lm(crim~dis, data = Boston)
crim.rad = lm(crim~rad, data = Boston)
crim.tax = lm(crim~tax, data = Boston)
crim.ptratio = lm(crim~ptratio, data = Boston)
crim.black = lm(crim~black, data = Boston)
crim.lstat = lm(crim~lstat, data = Boston)
crim.medv = lm(crim~medv, data = Boston)

summary(crim.zn)
```

```
##
## Call:
## lm(formula = crim ~ zn, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.429 -4.222 -2.620  1.250 84.523
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.45369    0.41722  10.675  < 2e-16 ***
## zn          -0.07393    0.01609  -4.594 5.51e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.435 on 504 degrees of freedom
## Multiple R-squared:  0.04019,    Adjusted R-squared:  0.03828
## F-statistic:  21.1 on 1 and 504 DF,  p-value: 5.506e-06
```

```r
summary(crim.indus)
```

```
##
## Call:
## lm(formula = crim ~ indus, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.972  -2.698  -0.736   0.712  81.813
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.06374    0.66723  -3.093  0.00209 **
## indus        0.50978    0.05102   9.991  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.866 on 504 degrees of freedom
## Multiple R-squared:  0.1653, Adjusted R-squared:  0.1637
## F-statistic: 99.82 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
summary(crim.chas)
```

```
##
## Call:
## lm(formula = crim ~ chas, data = Boston)
##
## 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
```

```r
summary(crim.nox)
```

```
##
## Call:
## lm(formula = crim ~ nox, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.371  -2.738  -0.974   0.559  81.728
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -13.720      1.699  -8.073 5.08e-15 ***
## nox           31.249      2.999  10.419  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.81 on 504 degrees of freedom
## Multiple R-squared:  0.1772, Adjusted R-squared:  0.1756
## F-statistic: 108.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
summary(crim.rm)
```

```
##
## Call:
## lm(formula = crim ~ rm, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
```

```
## -6.604 -3.952 -2.654  0.989 87.197
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   20.482      3.365   6.088 2.27e-09 ***
## rm            -2.684      0.532  -5.045 6.35e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.401 on 504 degrees of freedom
## Multiple R-squared:  0.04807,    Adjusted R-squared:  0.04618
## F-statistic: 25.45 on 1 and 504 DF,  p-value: 6.347e-07
```

```r
summary(crim.age)
```

```
##
## Call:
## lm(formula = crim ~ age, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.789 -4.257 -1.230  1.527 82.849
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.77791    0.94398  -4.002 7.22e-05 ***
## age          0.10779    0.01274   8.463 2.85e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.057 on 504 degrees of freedom
## Multiple R-squared:  0.1244, Adjusted R-squared:  0.1227
## F-statistic: 71.62 on 1 and 504 DF,  p-value: 2.855e-16
```

```r
summary(crim.dis)
```

```
##
## Call:
## lm(formula = crim ~ dis, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.708 -4.134 -1.527  1.516 81.674
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.4993     0.7304  13.006   <2e-16 ***
## dis          -1.5509     0.1683  -9.213   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.965 on 504 degrees of freedom
## Multiple R-squared:  0.1441, Adjusted R-squared:  0.1425
## F-statistic: 84.89 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
summary(crim.rad)
```

```
##
## Call:
## lm(formula = crim ~ rad, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.164  -1.381  -0.141   0.660  76.433
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.28716    0.44348  -5.157 3.61e-07 ***
## rad          0.61791    0.03433  17.998  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.718 on 504 degrees of freedom
## Multiple R-squared:  0.3913, Adjusted R-squared:   0.39
## F-statistic: 323.9 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
summary(crim.tax)
```

```
##
## Call:
## lm(formula = crim ~ tax, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.513  -2.738  -0.194   1.065  77.696
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.528369   0.815809  -10.45   <2e-16 ***
## tax          0.029742   0.001847   16.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.997 on 504 degrees of freedom
## Multiple R-squared:  0.3396, Adjusted R-squared:  0.3383
## F-statistic: 259.2 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
summary(crim.ptratio)
```

```
##
## Call:
## lm(formula = crim ~ ptratio, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.654 -3.985 -1.912  1.825 83.353
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.6469     3.1473  -5.607 3.40e-08 ***
```

```
## ptratio       1.1520     0.1694   6.801 2.94e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.24 on 504 degrees of freedom
## Multiple R-squared:  0.08407,    Adjusted R-squared:  0.08225
## F-statistic: 46.26 on 1 and 504 DF,  p-value: 2.943e-11
```

```r
summary(crim.black)
```

```
##
## Call:
## lm(formula = crim ~ black, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.756  -2.299  -2.095  -1.296  86.822
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.553529   1.425903  11.609   <2e-16 ***
## black       -0.036280   0.003873  -9.367   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.946 on 504 degrees of freedom
## Multiple R-squared:  0.1483, Adjusted R-squared:  0.1466
## F-statistic: 87.74 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
summary(crim.lstat)
```

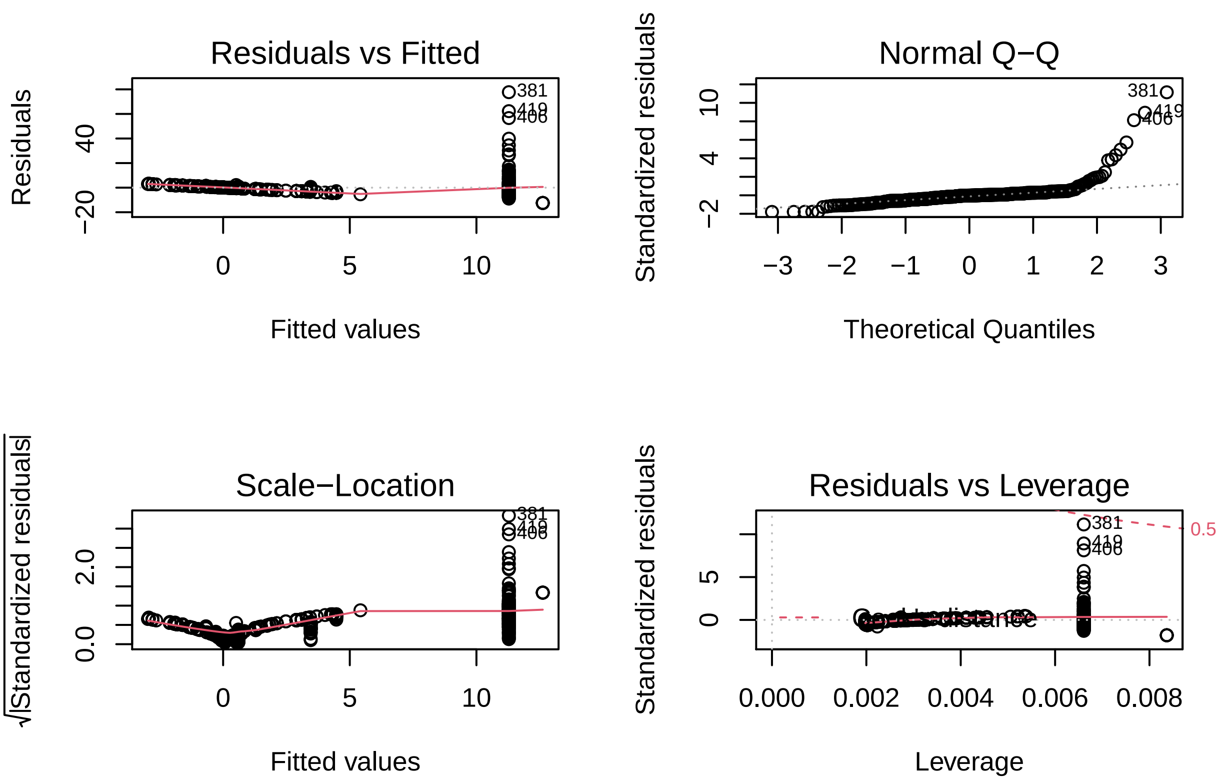
```
##
## Call:
## lm(formula = crim ~ lstat, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.925  -2.822  -0.664   1.079  82.862
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.33054    0.69376  -4.801 2.09e-06 ***
## lstat        0.54880    0.04776  11.491  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.664 on 504 degrees of freedom
## Multiple R-squared:  0.2076, Adjusted R-squared:  0.206
## F-statistic:   132 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
summary(crim.medv)
```

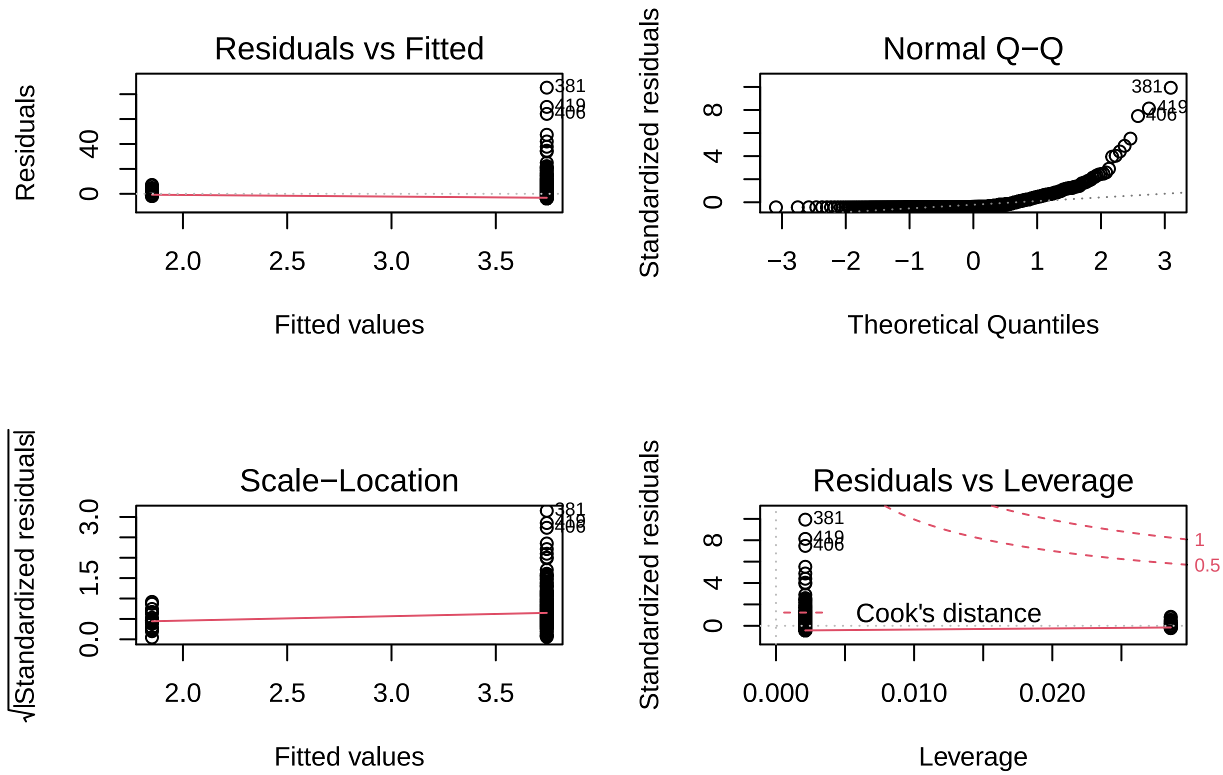
```
##
## Call:
## lm(formula = crim ~ medv, data = Boston)
##
```

```
## Residuals:
##     Min      1Q Median      3Q     Max
## -9.071 -4.022 -2.343  1.298 80.957
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.79654    0.93419   12.63   <2e-16 ***
## medv        -0.36316    0.03839   -9.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.934 on 504 degrees of freedom
## Multiple R-squared:  0.1508, Adjusted R-squared:  0.1491
## F-statistic: 89.49 on 1 and 504 DF,  p-value: < 2.2e-16
```

```r
par(mfrow = c(2,2))

plot(crim.tax)
```




```r
plot(crim.chas)
```



All of the linear models show statistical significance between crime and the corresponding x variable, except for crime and chas. The p-value for chas with respect to crim was 0.209. For all of the other plots, the p-values were statistically significant.

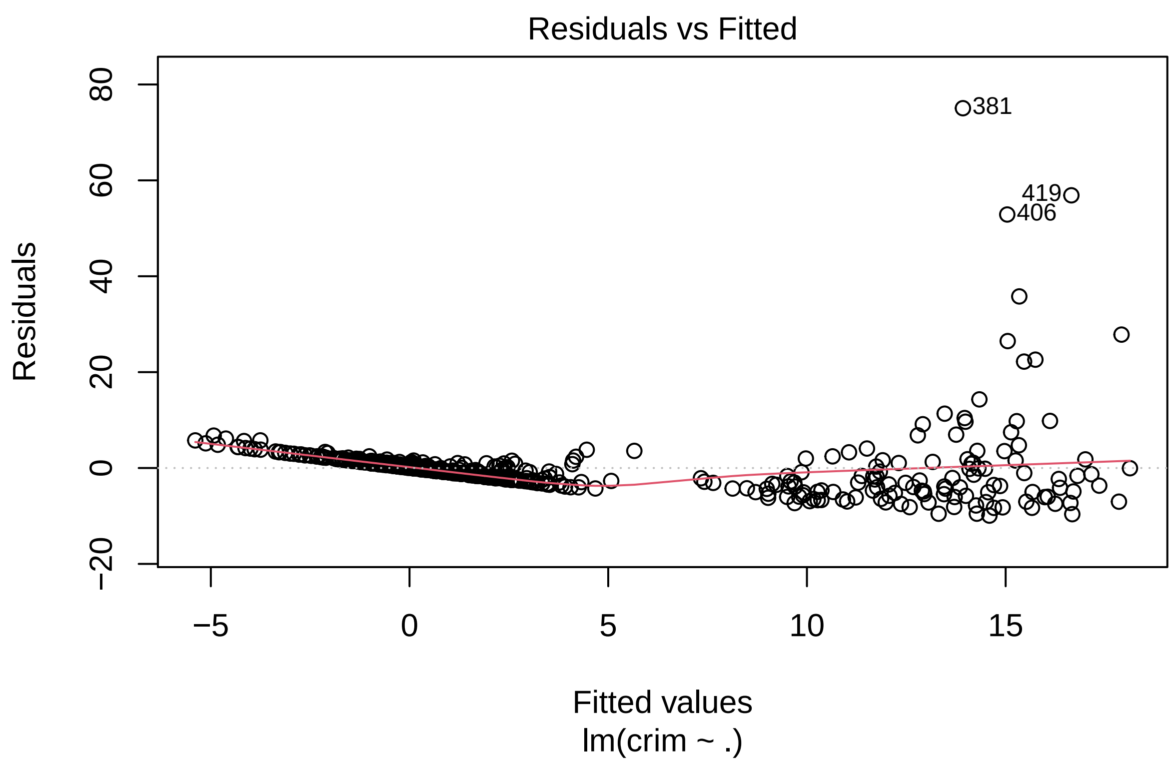
Chapter 3 - 15b

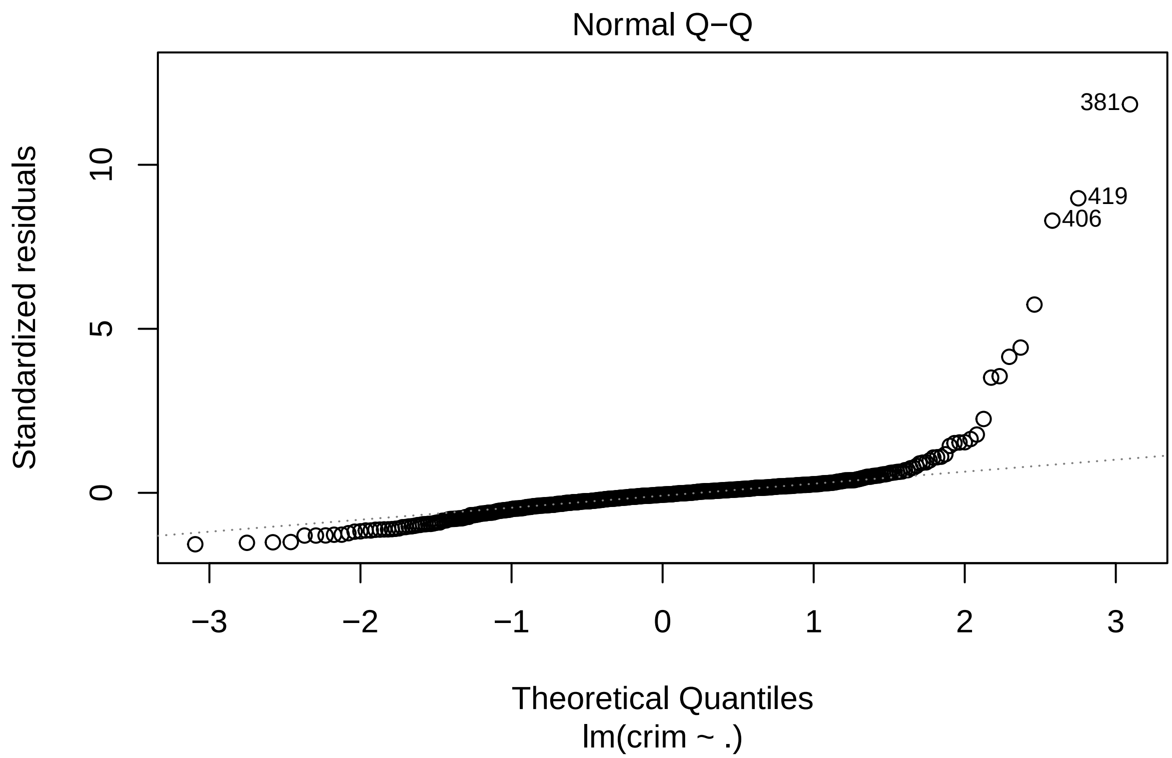
```
crim.all = lm(crim~., data = Boston)

summary(crim.all)
```

```
##
## 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
```

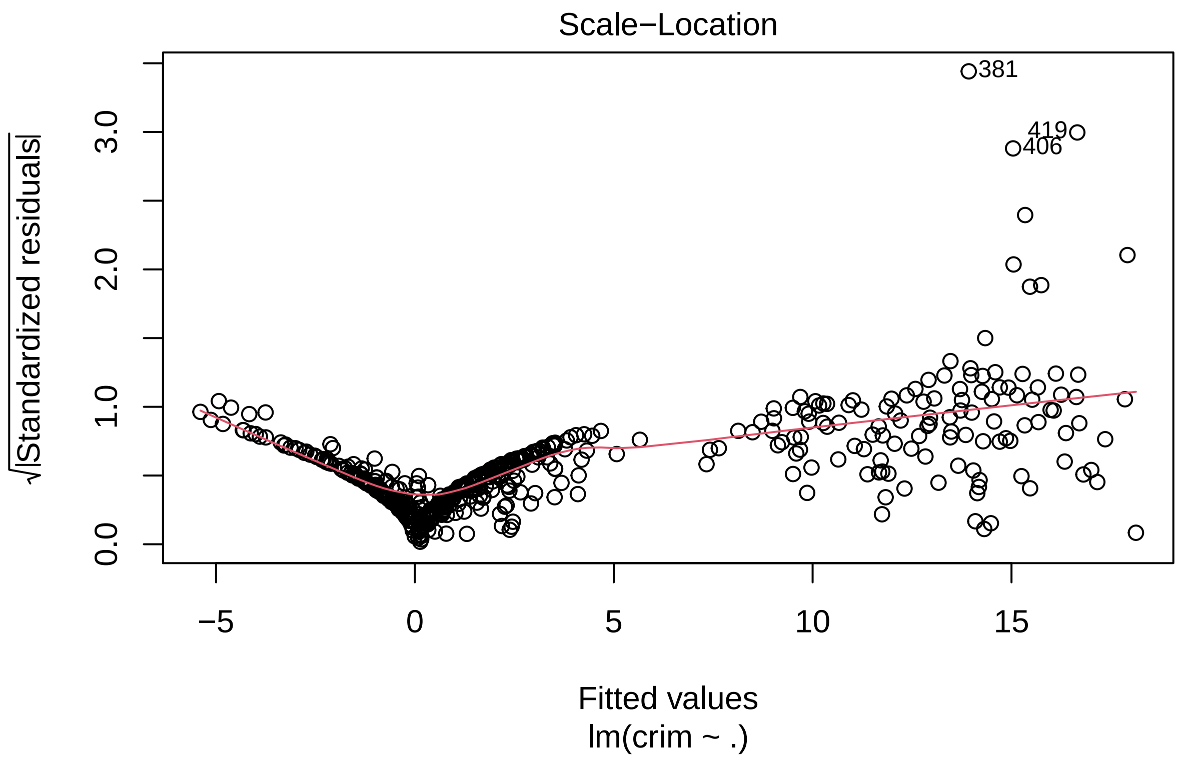
```
## dis          -0.987176   0.281817  -3.503 0.000502 ***
## rad           0.588209   0.088049   6.680 6.46e-11 ***
## tax          -0.003780   0.005156  -0.733 0.463793
## ptratio      -0.271081   0.186450  -1.454 0.146611
## black        -0.007538   0.003673  -2.052 0.040702 *
## lstat         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
```

```r
plot(crim.all)
```

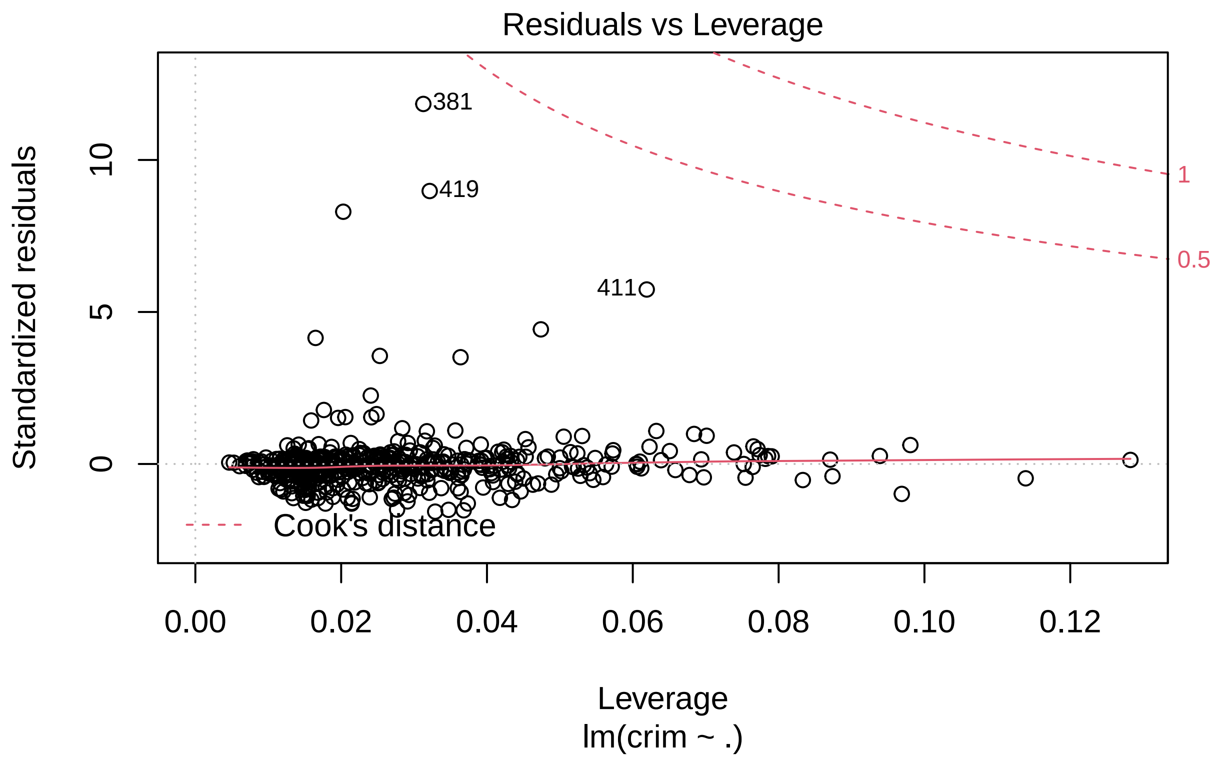




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(crim ~ .)
381
419
406
10
5
0
−3
−2
−1
0
1
2
3



Scale−Location
√|Standardized residuals|
0.0
1.0
2.0
3.0
381
419
406
−5
0
5
10
15
Fitted values
lm(crim ~ .)



Rejecting the null hypothesis is reserved for the predictors which exhibit statistical significance. Those predictors must have p-values less than 0.05. We can reject the null hypothesis for zn, dis, rad, black, and medv.

Chapter 3 - 15c

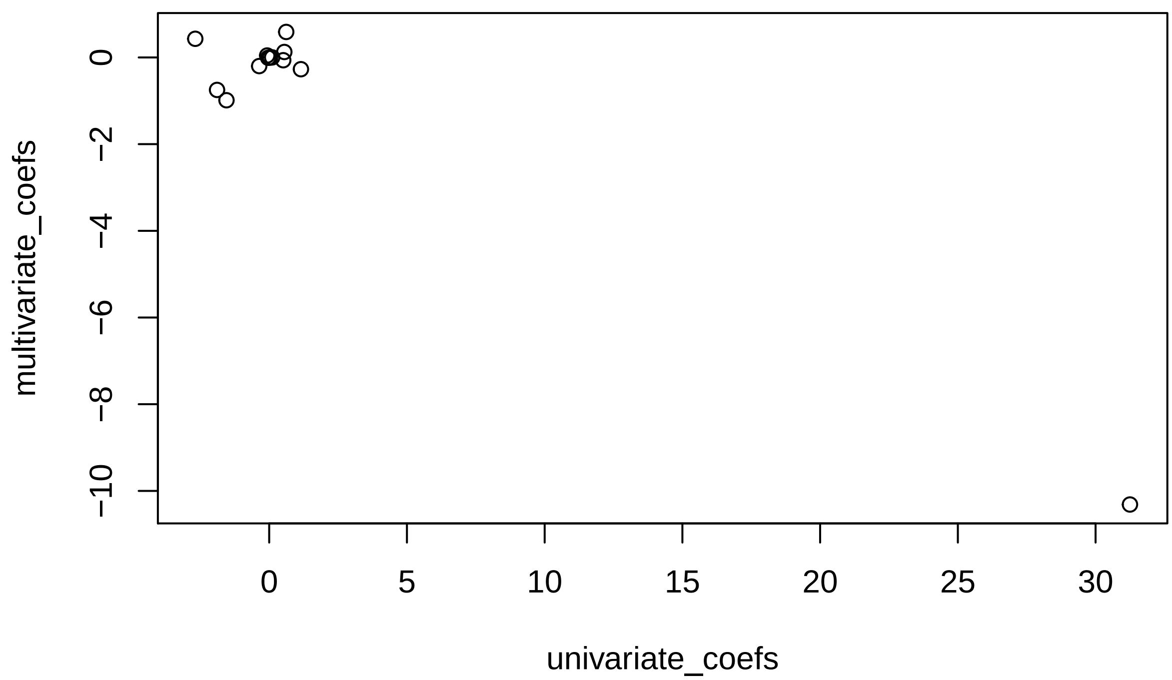
```
univariate_coefs <- c(coef(crim.zn)[2], coef(crim.indus)[2], coef(crim.chas)[2], coef(crim.nox)[2], coe
                      coef(crim.age)[2], coef(crim.dis)[2], coef(crim.rad)[2], coef(crim.tax)[2], coef(
                      coef(crim.black)[2], coef(crim.lstat)[2], coef(crim.medv)[2])
univariate_coefs
```

```
##          zn       indus        chas         nox          rm         age
## -0.07393498  0.50977633 -1.89277655 31.24853120 -2.68405122  0.10778623
##         dis         rad         tax     ptratio       black       lstat
## -1.55090168  0.61791093  0.02974225  1.15198279 -0.03627964  0.54880478
##        medv
## -0.36315992
```

```
multivariate_coefs <- coef(crim.all)[2:14]
multivariate_coefs
```

```
##            zn         indus          chas           nox            rm
##   0.044855215  -0.063854824  -0.749133611 -10.313534912   0.430130506
##           age           dis           rad           tax       ptratio
##   0.001451643  -0.987175726   0.588208591  -0.003780016  -0.271080558
##         black         lstat          medv
##  -0.007537505   0.126211376  -0.198886821
```

```r
par(mfrow = c(1,1))

plot(univariate_coefs, multivariate_coefs)
```



The results from part a vary with those from part b. For example, the univariate coefficient for rad is 0.6179, while the multivariate coefficient for rad is 0.5882. This doesn't show much difference. However the univariate coefficient for nox shows a positive correlation of 31.2485, while the multivariate coefficient for nox is -10.3135. This shows a large amount of difference. It implies that nox is correlated positively with crim when used as a single predictor, but when combined with more attributes, the contribution of nox becomes a huge negative correlation.

Chapter 3 - 15d

```r
crim.zn2 = lm(crim~poly(zn,3), data = Boston)
crim.indus2 = lm(crim~poly(indus,3), data = Boston)
crim.nox2 = lm(crim~poly(nox,3), data = Boston)
crim.rm2 = lm(crim~poly(rm,3), data = Boston)
crim.age2 = lm(crim~poly(age,3), data = Boston)
crim.dis2 = lm(crim~poly(dis,3), data = Boston)
crim.rad2 = lm(crim~poly(rad,3), data = Boston)
crim.tax2 = lm(crim~poly(tax,3), data = Boston)
crim.ptratio2 = lm(crim~poly(ptratio,3), data = Boston)
crim.black2 = lm(crim~poly(black,3), data = Boston)
crim.lstat2 = lm(crim~poly(lstat,3), data = Boston)
crim.medv2 = lm(crim~poly(medv,3), data = Boston)
```

```r
summary(crim.zn2)
```

```
##
## Call:
## lm(formula = crim ~ poly(zn, 3), data = Boston)
##
## 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
```

```r
summary(crim.indus2)
```

```
##
## Call:
## lm(formula = crim ~ poly(indus, 3), data = Boston)
##
## 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
```

```r
summary(crim.nox2)
```

```
##
## Call:
## lm(formula = crim ~ poly(nox, 3), data = Boston)
##
## 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
```

```r
summary(crim.rm2)
```

```
##
## Call:
## lm(formula = crim ~ poly(rm, 3), data = Boston)
##
## 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
```

```r
summary(crim.age2)
```

```
##
## Call:
## lm(formula = crim ~ poly(age, 3), data = Boston)
##
## 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
```

```r
summary(crim.dis2)
```

```
##
## Call:
## lm(formula = crim ~ poly(dis, 3), data = Boston)
##
## 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
```

```r
summary(crim.rad2)
```

```
##
## Call:
## lm(formula = crim ~ poly(rad, 3), data = Boston)
##
## 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
```

```r
summary(crim.tax2)
```

```
##
## Call:
## lm(formula = crim ~ poly(tax, 3), data = Boston)
```

```
##
## 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
```

```r
summary(crim.ptratio2)
```

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

```r
summary(crim.black2)
```

```
##
## Call:
## lm(formula = crim ~ poly(black, 3), data = Boston)
##
## 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
```

```r
summary(crim.lstat2)
```

```
##
## Call:
## lm(formula = crim ~ poly(lstat, 3), data = Boston)
##
## 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
```

```r
summary(crim.medv2)
```

```
##
## Call:
## lm(formula = crim ~ poly(medv, 3), data = Boston)
##
## 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
```

The cubic coefficient for zn, rm, rad, tax, black, and lstat are not significant. It was not possible to find a polynomial regression between crim and chas because there were only 2 unique values for chas: 0 and 1. The only quadratic coefficient which did not show significance was for black. Therefore the relationship between crim and black can only be explained somewhat by a linear regression. Many of these variables can be explained by a cubic or quadratic equation, but that might be due to over-fitting and not necessarily indicative of a real pattern.

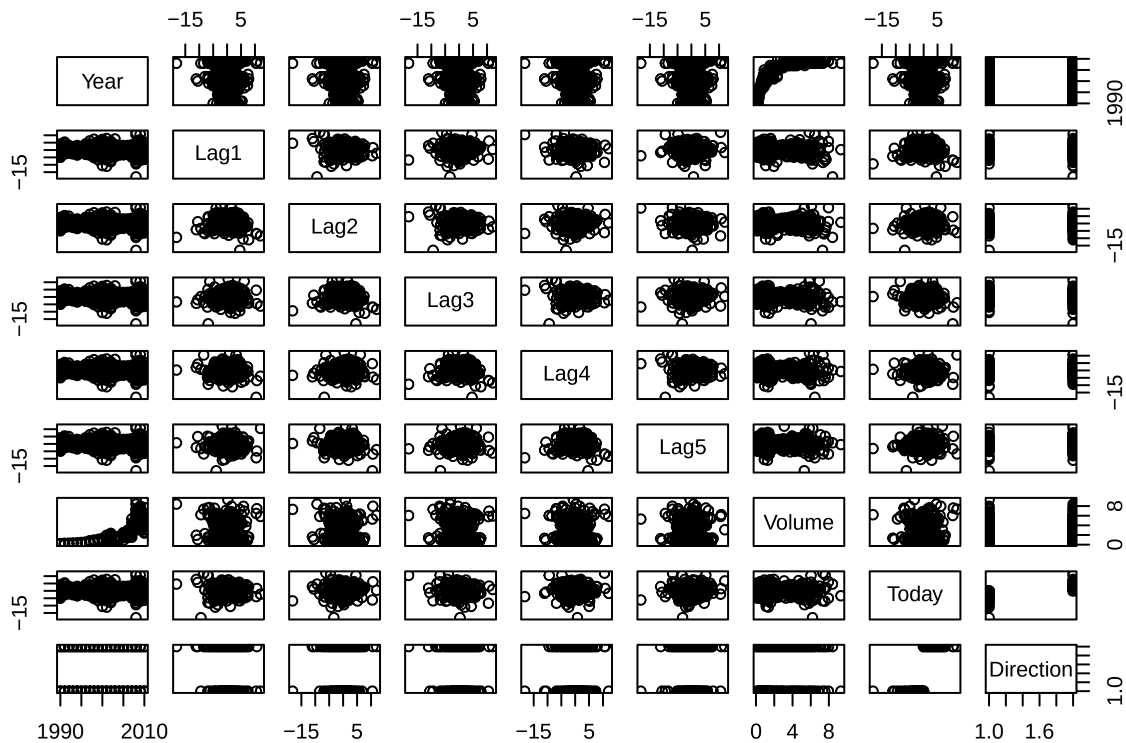
Chapter 4 - 10

Chapter 4 - 10a

```
rm(list = ls())
library(ISLR)

summary(Weekly)
```

```
##       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            Today
##  Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747   Min.   :-18.1950
##  1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202   1st Qu.: -1.1540
##  Median :  0.2380   Median :  0.2340   Median :1.00268   Median :  0.2410
##  Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462   Mean   :  0.1499
##  3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373   3rd Qu.:  1.4050
##  Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821   Max.   : 12.0260
##  Direction
##  Down:484
##  Up  :605
##
##
##
##
```

```
pairs(Weekly)
```




```r
cor(Weekly[,-9])
```

```
##               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
```

There appears to be a non-linear positive relationship between Year and Volume.The correlation between the other variables looks to be small and not discernible.

Chapter 4 - 10b

```r
attach(Weekly)

glm.fit = glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume, data=Weekly, family=binomial)

summary(glm.fit)
```

```
##
## 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
```

The only predictor that appears to be statistically significant is the Lag2 variable with a p-value of 0.0296.

Chapter 4 - 10c

```r
glm.probs = predict(glm.fit, type = 'response')
contrasts(Direction)
```

```
##      Up
## Down  0
## Up    1
```

```r
glm.pred = rep("Down", length(glm.probs))
glm.pred[glm.probs > 0.5] = "Up"

table(glm.pred, Direction)
```

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

```
##     Up    430 557
```

```r
mean(glm.pred == Direction)
```

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

```r
down_correct <- 54 / (54 + 430)
down_correct
```

```
## [1] 0.1115702
```

```r
up_correct <- 557 / (557 + 48)
up_correct
```

```
## [1] 0.9206612
```

Fraction of correct prediction is (54 + 557) / 1089 which equals 0.561. When the logistic regression is trying to predict when the market is up, it is correct 92% of the time. When the logistic regression is trying to predict when the market is down, it is correct only 11% of the time.

Chapter 4 - 10d

```r
train = (Year < 2009)
glm.fit2 = glm(Direction~Lag2, data = Weekly, family = binomial, subset = train)
test = Weekly[!train,]
glm.probs2 = predict(glm.fit2, test, type = 'response')
glm.pred2 = rep("Down", length(glm.probs2))
glm.pred2[glm.probs2 > 0.5] = "Up"
Direction.test = Direction[!train]

table(glm.pred2, Direction.test)
```

```
##          Direction.test
## glm.pred2 Down Up
##      Down    9  5
##      Up     34 56
```

```r
mean(glm.pred2 == Direction.test)
```

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

The overall fraction of correct predictions on the test data set is 0.625 or 62.5%.

Chapter 4 - 10g

```r
library(class)
train.X = matrix((Lag2)[train])
test.X = matrix((Lag2)[!train])
train.Direction = Direction[train]

set.seed(1)
knn.pred = knn(train.X, test.X, train.Direction, k = 1)

table(knn.pred, Direction.test)
```

```
##         Direction.test
## knn.pred Down Up
##     Down   21 30
```

```
##     Up     22 31
```

```r
mean(knn.pred == Direction.test)
```

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

The overall fraction of correct predictions on the test data set is 0.5 or 50%.

Chapter 4 - 10h

The logistic regression appears to provide the best predictions because it correctly predicted 62.5% on the test data set, while the KNN method only predicted 50% on the test data set.

Chapter 4 - 10i

```r
glm.fit3 = glm(Direction~Lag3 + Lag4 + Lag5, data = Weekly, family = binomial, subset = train)
glm.probs3 = predict(glm.fit3, test, type = 'response')
glm.pred3 = rep("Down", length(glm.probs3))
glm.pred3[glm.probs3 > 0.5] = "Up"

table(glm.pred3, Direction.test)
```

```
##          Direction.test
## glm.pred3 Down Up
##      Down    0  3
##      Up     43 58
```

```r
mean(glm.pred3 == Direction.test)
```

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

```r
set.seed(1)
knn.pred2 = knn(train.X, test.X, train.Direction, k = 20)

table(knn.pred2, Direction.test)
```

```
##          Direction.test
## knn.pred2 Down Up
##      Down   21 21
##      Up     22 40
```

```r
mean(knn.pred2 == Direction.test)
```

```
## [1] 0.5865385
```

```r
set.seed(1)
knn.pred3 = knn(train.X, test.X, train.Direction, k = 100)

table(knn.pred3, Direction.test)
```

```
##          Direction.test
## knn.pred3 Down Up
##      Down   10 11
##      Up     33 50
```

```r
mean(knn.pred3 == Direction.test)
```

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

```r
glm.fit4 = glm(Direction~Lag1 + Lag2 + Lag3 + Lag4 + Lag5, data = Weekly, family = binomial, subset = t
glm.probs4 = predict(glm.fit4, test, type = 'response')
glm.pred4 = rep("Down", length(glm.probs4))
glm.pred4[glm.probs4 > 0.5] = "Up"

table(glm.pred4, Direction.test)
```

```
##          Direction.test
## glm.pred4 Down Up
##      Down   10 14
##      Up     33 47
```

```r
mean(glm.pred4 == Direction.test)
```

```
## [1] 0.5480769
```

Logistic regression with Lag3, Lag4, and Lag5 as predictors yields a correct prediction rate of 55.8%. KNN method with k = 20 yields a correct prediction rate of 56.65%. KNN method with k = 100 yields a correct prediction rate of 57.7%. Logistic regression with all Lag variables as predictors yields a correct prediction rate of 54.8%. The original logistic regression with Lag2 as the only predictor yielded the best prediction rate.

Chapter 6 - 9

Chapter 6 - 9a

```r
rm(list = ls())
library(ISLR)

set.seed(1)
train = sample(c(TRUE, FALSE), nrow(College), rep=TRUE)
test=(!train)
College_train = College[train,]
College_test = College[-train,]
```

Chapter 6 - 9b

```r
colnames(College)
```

```
##  [1] "Private"     "Apps"        "Accept"      "Enroll"      "Top10perc"
##  [6] "Top25perc"   "F.Undergrad" "P.Undergrad" "Outstate"    "Room.Board"
## [11] "Books"       "Personal"    "PhD"         "Terminal"    "S.F.Ratio"
## [16] "perc.alumni" "Expend"      "Grad.Rate"
```

```r
set.seed(1)

lm.fit = lm(Apps~., data = College_train)
lm.pred = predict(lm.fit, College_test)

Mean_squares <- mean((lm.pred - College_test$Apps)^2)

RMSE_apps <- sqrt(Mean_squares)

RMSE_apps
```

```
## [1] 1050.898
```

RMSE of the least squares model on the training set is 1050.898.

Chapter 6 - 9c

```
library(glmnet)
```

```
## Loading required package: Matrix
## Loaded glmnet 4.0-2
```

```
set.seed(1)

grid = 10 ^ seq(10, -2, length = 100)
x_train = model.matrix(Apps~., College_train)[,-1]
x_test = model.matrix(Apps~., College_test)[,-1]
y_train = College$Apps[train]
y_test = College$Apps[-train]
ridge.mod = cv.glmnet(x_train, y_train, alpha = 0)
lambda_min <- ridge.mod$lambda.min
lambda_min
```

```
## [1] 394.2365
```

```
ridge.pred = predict(ridge.mod, newx = x_test, s = lambda_min)

sqrt(mean((ridge.pred - y_test)^2))
```

```
## [1] 1126.992
```

RMSE of the ridge regression is 1126.992.

Chapter 6 - 9d

```
set.seed(1)

lasso.mod = cv.glmnet(x_train, y_train, alpha = 1)
lambda_min2 <- lasso.mod$lambda.min
lambda_min2
```

```
## [1] 59.92044
```

```
lasso.pred = predict(lasso.mod, newx = x_test, s = lambda_min2)

sqrt(mean((lasso.pred - y_test)^2))
```

```
## [1] 1092
```

```
lasso.mod2 = glmnet(model.matrix(Apps~., data = College), College[, "Apps"], alpha = 1)

lasso_coef = predict(lasso.mod2, s = lambda_min2, type = "coefficients")

lasso_coef
```

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                         1
## (Intercept) -778.60692783
## (Intercept)    .
## PrivateYes  -289.87184550
```

```
## Accept         1.39119211
## Enroll         .
## Top10perc     25.96539500
## Top25perc      .
## F.Undergrad    .
## P.Undergrad    .
## Outstate      -0.01184841
## Room.Board     0.04090173
## Books          .
## Personal       .
## PhD           -0.39507515
## Terminal      -0.89365231
## S.F.Ratio      .
## perc.alumni    .
## Expend         0.05095596
## Grad.Rate      0.30670768
```

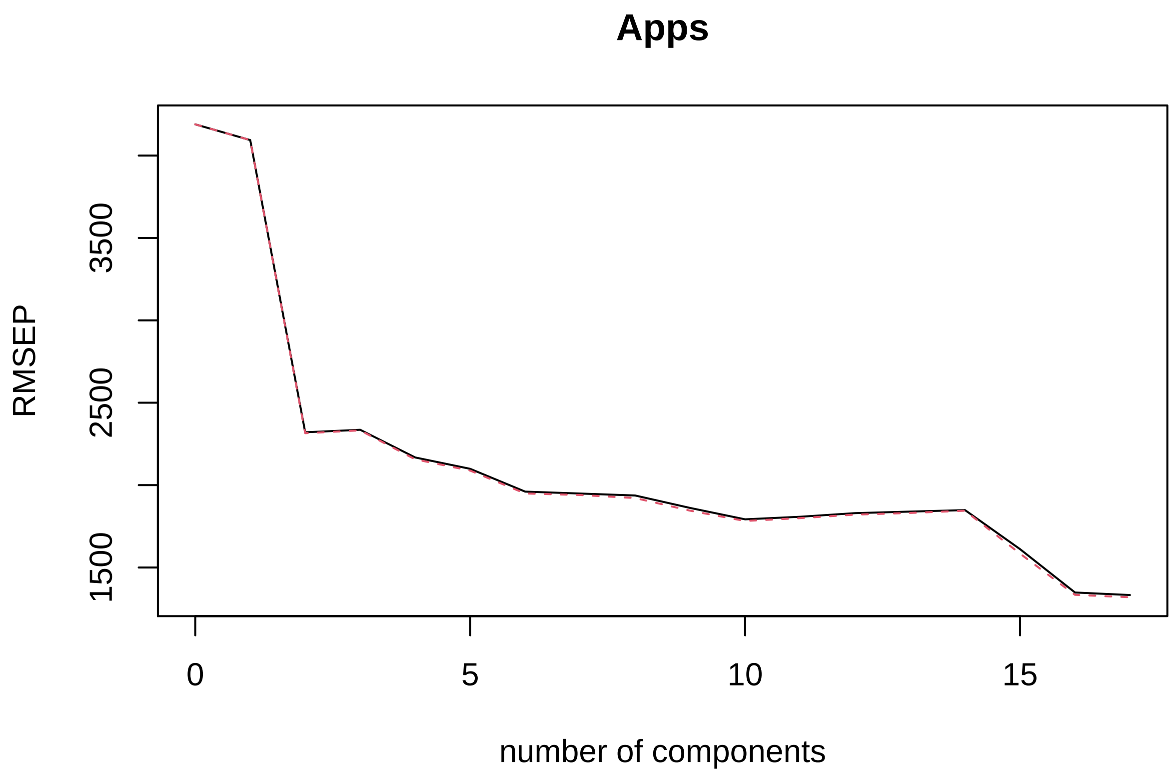
RMSE of the lasso regression is 1088.011. The non-zero coefficients are PrivateYes, Accept, Top10perc, Outstate, Room.Board, PhD, Terminal, perc.alumni, Expend, and Grad.Rate.

Chapter 6 - 9e

```r
library(pls)
```

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

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

```r
set.seed(1)

pcr.fit = pcr(Apps~., data = College_train, scale = TRUE, validation = "CV")

par(mfrow = c(1,1))

validationplot(pcr.fit, val.type = "RMSEP")
```




```r
summary(pcr.fit)
```

```
## Data:    X dimension: 393 17
##  Y dimension: 393 1
## Fit method: svdpc
## Number of components considered: 17
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV            4189     4094     2321     2336     2168     2099     1961
## adjCV         4189     4094     2315     2332     2156     2088     1949
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV        1949     1937     1861      1792      1808      1830      1839
## adjCV     1940     1922     1845      1783      1800      1821      1831
##        14 comps  15 comps  16 comps  17 comps
## CV         1848      1612      1348      1333
## adjCV      1845      1584      1334      1320
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X      31.858    57.44    64.20    69.91    75.10    80.17    83.82    87.30
## Apps    4.353    70.99    71.18    76.84    78.34    81.03    81.59    82.21
##       9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
## X       90.26     92.74     94.79     96.70     97.76     98.67     99.37
## Apps    83.31     83.97     83.97     84.34     84.58     84.70     91.28
```

```
##       16 comps  17 comps
## X        99.82    100.00
## Apps     92.83     93.02
```

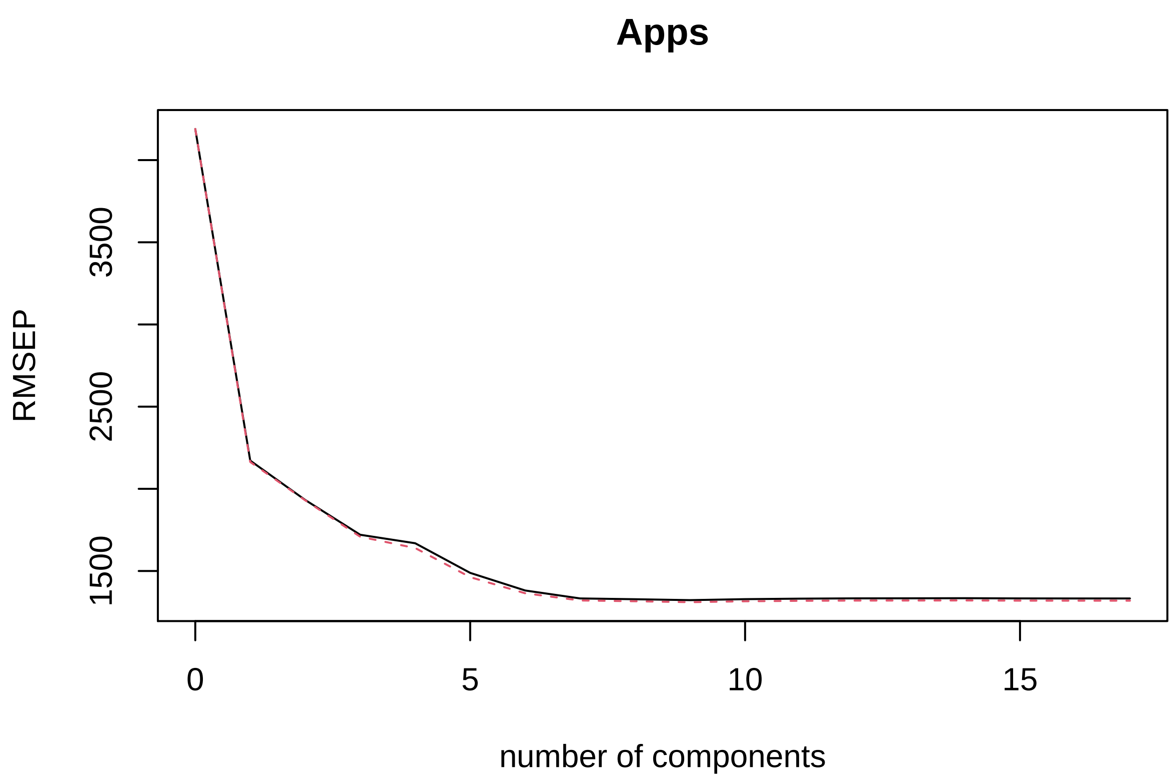
```r
pcr.pred = predict(pcr.fit, College_test, ncomp = 17)

sqrt(mean((pcr.pred - y_test)^2))
```

```
## [1] 1050.898
```

Minimum RMSE for PCR occurred at M = 17, which produced an RMSE of 1051.

Chapter 6 - 9f

```r
set.seed(1)

pls.fit = plsr(Apps~., data = College_train, scale = TRUE, validation = "CV")

validationplot(pls.fit, val.type = "RMSEP")
```




```r
summary(pls.fit)
```

```
## Data:    X dimension: 393 17
##  Y dimension: 393 1
## Fit method: kernelpls
## Number of components considered: 17
##
## VALIDATION: RMSEP
```

```
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV            4189     2172     1932     1720     1669     1489     1382
## adjCV         4189     2163     1930     1709     1640     1463     1365
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV        1333     1328     1323      1329      1332      1334      1334
## adjCV     1321     1316     1310      1316      1319      1320      1321
##        14 comps  15 comps  16 comps  17 comps
## CV         1335      1333      1333      1333
## adjCV      1321      1320      1320      1320
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X       26.01    44.96    62.49    65.22    68.52    72.89    77.13    80.46
## Apps    75.74    82.40    86.74    90.58    92.34    92.79    92.88    92.93
##       9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
## X       82.45     84.76     88.08     90.76     92.80     94.45     97.02
## Apps    92.98     93.00     93.01     93.01     93.02     93.02     93.02
##       16 comps  17 comps
## X        98.03    100.00
## Apps     93.02     93.02
```

```r
pls.pred = predict(pls.fit, College_test, ncomp = 10)

sqrt(mean((pls.pred - y_test)^2))
```

```
## [1] 1054.069
```

Minimum RMSE for PLS occurred at M = 10, which produced an RMSE of 1054.

Chapter 6 - 9g

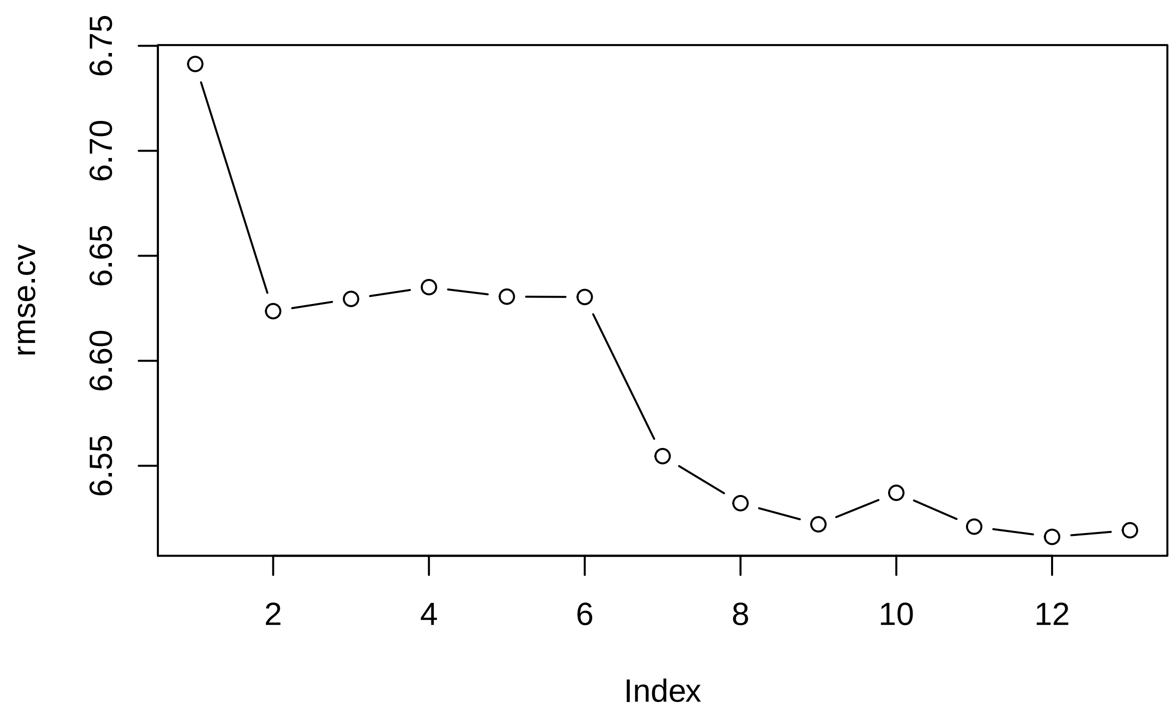
The RMSE of the least squares model was 1051. The RMSE of the ridge regression was 1127. The RMSE of the lasso regression was 1088. The RMSE of the PCR model was 1051. The RMSE of the PLS model was 1054. It seems that the least squares and the PCR models predicted the number of college applications the most accurately. However. there doesn't appear to be too much difference in the methods used.

Chapter 6 - 11

Chapter 6 - 11a

```r
rm(list = ls())

set.seed(1)
library(MASS)
library(glmnet)
library(leaps)

predict.regsubsets = function(object, newdata, id, ...) {
  form = as.formula(object$call[[2]])
  mat = model.matrix(form, newdata)
  coefi = coef(object, id = id)
  xvars = names(coefi)
  mat[, names(coefi)] %*% coefi
}
```

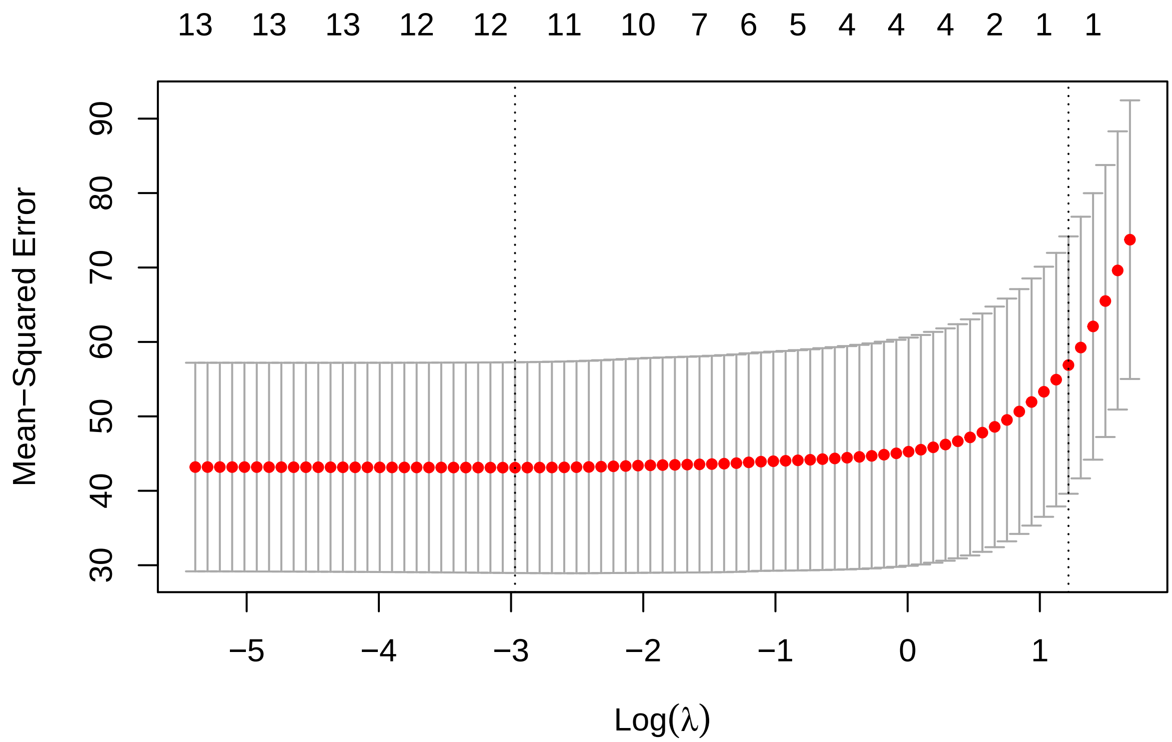
```
k = 10
folds = sample(1:k, nrow(Boston), replace = TRUE)
cv.errors = matrix(NA, k, 13, dimnames = list(NULL, paste(1:13)))
for (j in 1:k) {
  best.fit=regsubsets(crim~.,data = Boston [folds != j,], nvmax = 13)
  for(i in 1:13) {
    pred = predict(best.fit,Boston[folds == j,],id = i)
    cv.errors[j,i] = mean( (Boston$crim[folds == j]-pred)^2)
  }
}
rmse.cv = sqrt(apply(cv.errors, 2, mean))
plot(rmse.cv, type = "b")
```




```
a <- which.min(rmse.cv)
rmse.cv[a]
```

```
##       12
## 6.516145
```

```
x = model.matrix(crim~. -1, data = Boston)
y = Boston$crim
cv.lasso = cv.glmnet(x, y, type.measure = "mse")
plot(cv.lasso)
```




```r
coef(cv.lasso)
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                     1
## (Intercept) 1.4186415
## zn          .
## indus       .
## chas        .
## nox         .
## rm          .
## age         .
## dis         .
## rad         0.2298449
## tax         .
## ptratio     .
## black       .
## lstat       .
## medv        .
```

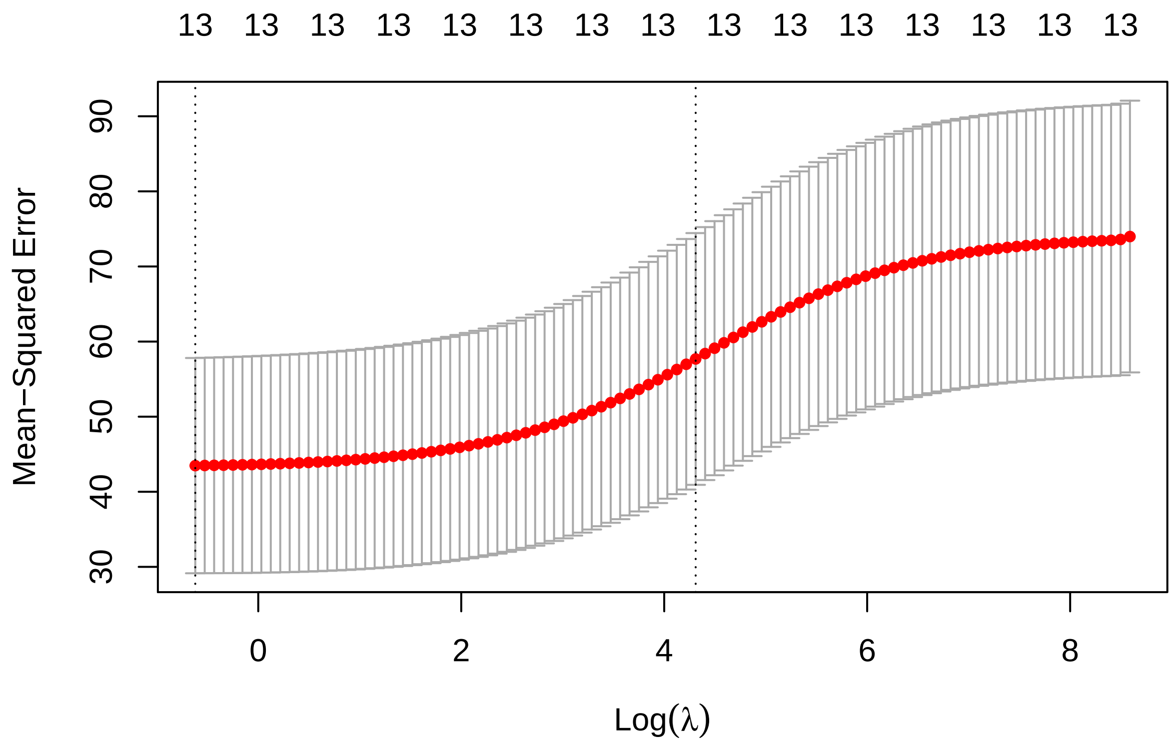
```r
lasso.min = cv.lasso$lambda.min
lasso.min
```

```
## [1] 0.0513069
```

```r
sqrt(cv.lasso$cvm[cv.lasso$lambda == cv.lasso$lambda.1se])
```

```
## [1] 7.542567
```

```r
cv.ridge = cv.glmnet(x, y, type.measure = "mse", alpha = 0)
plot(cv.ridge)
```




```r
coef(cv.ridge)
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                        1
## (Intercept)  1.400465359
## zn          -0.002958004
## indus        0.030629045
## chas        -0.179936009
## nox          1.970152366
## rm          -0.148397786
## age          0.006502245
## dis         -0.099606703
## rad          0.049161233
## tax          0.002222499
## ptratio      0.075064827
## black       -0.002779273
## lstat        0.037863265
## medv        -0.024852582
```

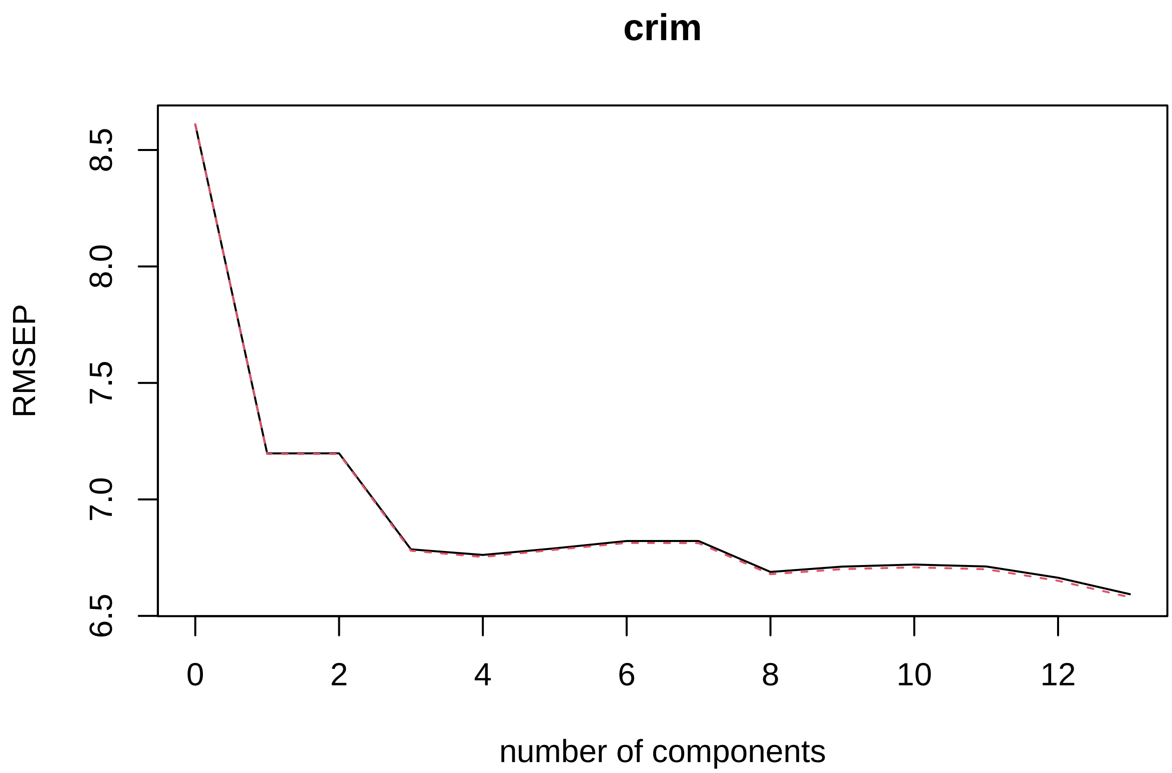
```r
sqrt(cv.ridge$cvm[cv.ridge$lambda == cv.ridge$lambda.1se])
```

```
## [1] 7.595074
```

```r
library(pls)

pcr.fit = pcr(crim ~ ., data = Boston, scale = TRUE, validation = "CV")
summary(pcr.fit)
```

```
## Data:    X dimension: 506 13
##  Y dimension: 506 1
## Fit method: svdpc
## Number of components considered: 13
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV            8.61    7.198    7.198    6.786    6.762    6.790    6.821
## adjCV         8.61    7.195    7.195    6.780    6.753    6.784    6.813
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV       6.822    6.689    6.712     6.720     6.712     6.664     6.593
## adjCV    6.812    6.679    6.701     6.708     6.700     6.651     6.580
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X       47.70    60.36    69.67    76.45    82.99    88.00    91.14    93.45
## crim    30.69    30.87    39.27    39.61    39.61    39.86    40.14    42.47
##       9 comps  10 comps  11 comps  12 comps  13 comps
## X       95.40     97.04     98.46     99.52     100.0
## crim    42.55     42.78     43.04     44.13      45.4
```

```r
validationplot(pcr.fit, val.type = "RMSEP")
```



Best subset selection produced an RMSE of 6.52 Lasso regression produced an RMSE of 7.54. Ridge regression produced an RMSE of 7.595. 13 component PCR fit has the lowest RMSE.

Chapter 6 - 11b

I would recommend the best subset selection as it gave the lowest RMSE.

Chapter 6 - 11c

The best subset selection does include all variables in the data set.

Chapter 8 - 8

Chapter 8 - 8a

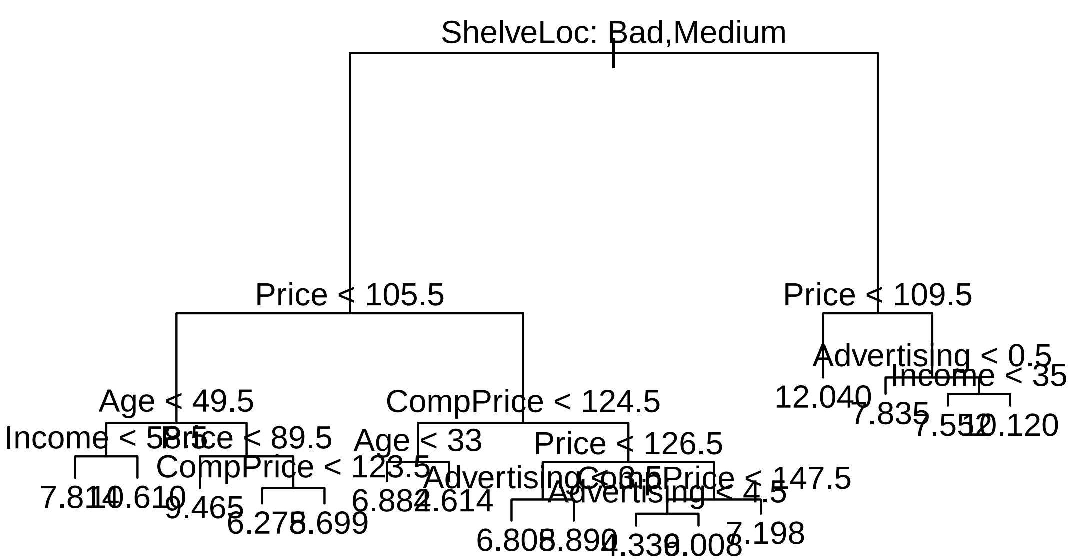
```
rm(list = ls())

library(ISLR)
attach(Carseats)
set.seed(1)

train = sample(dim(Carseats)[1], 0.8 * dim(Carseats)[1])
Carseats_train = Carseats[train,]
Carseats_test = Carseats[-train,]
```

Chapter 8 - 8b

```r
library(tree)

tree_carseats = tree(Sales~., data = Carseats_train)

summary(tree_carseats)
```

```
##
## Regression tree:
## tree(formula = Sales ~ ., data = Carseats_train)
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "Age"         "Income"      "CompPrice"
## [6] "Advertising"
## Number of terminal nodes:  16
## Residual mean deviance:  2.572 = 781.9 / 304
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -4.45400 -1.07000 -0.05544  0.00000  1.14500  4.69600
```

```r
plot(tree_carseats)
text(tree_carseats, pretty = 0)
```



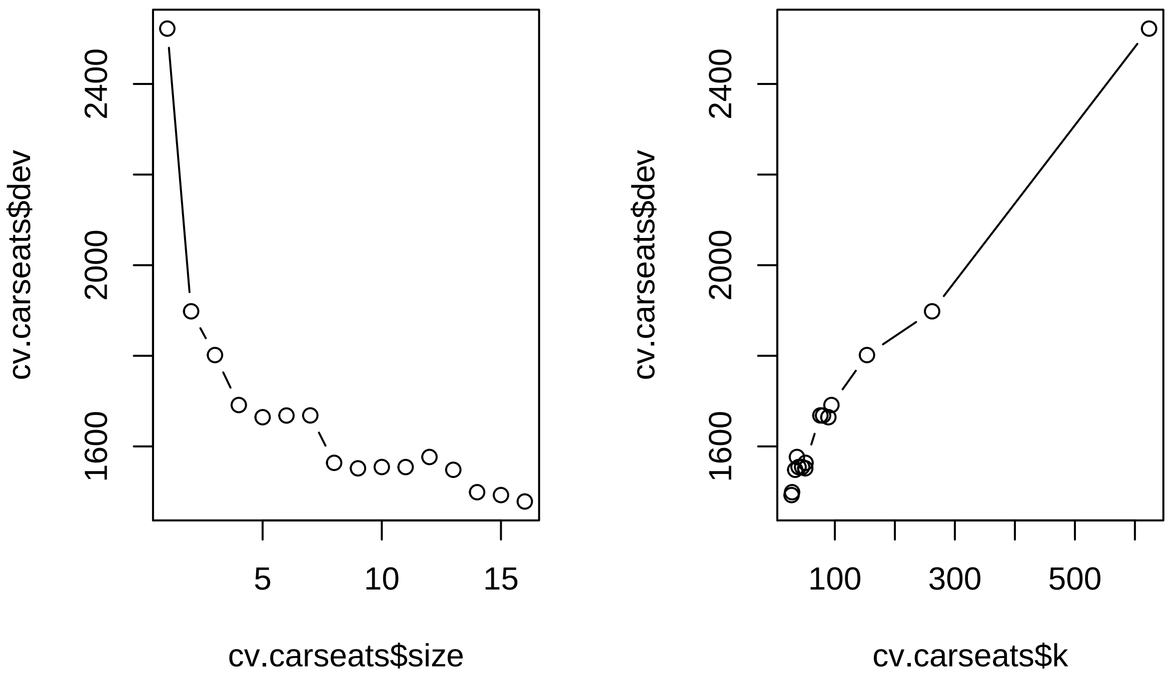

```r
pred_carseats = predict(tree_carseats, Carseats_test)
mean((Carseats_test$Sales - pred_carseats)^2)
```

```
## [1] 4.936081
```

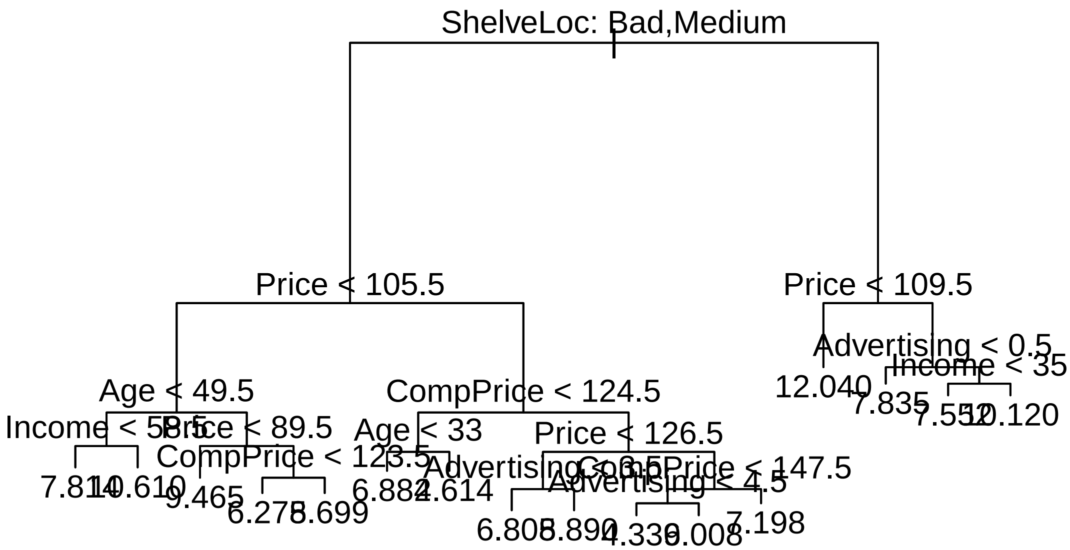
The test MSE is approximately 4.94.

Chapter 8 - 8c

```
cv.carseats = cv.tree(tree_carseats, FUN = prune.tree)
par(mfrow = c(1,2))
plot(cv.carseats$size, cv.carseats$dev, type = "b")
plot(cv.carseats$k, cv.carseats$dev, type = "b")
```




```
prune_carseats = prune.tree(tree_carseats, best = 16)
par(mfrow = c(1,1))
plot(prune_carseats)
text(prune_carseats, pretty = 0)
```




```r
pred_pruned = predict(prune_carseats, Carseats_test)
mean((Carseats_test$Sales - pred_pruned)^2)
```

```
## [1] 4.936081
```

Pruning the tree does not affect the RMSE very much as it is still approximately 4.94.

Chapter 8 - 8d

```r
library(randomForest)
```

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

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

```r
bag_carseats = randomForest(Sales~., data = Carseats_train, mtry = 10, ntree = 500, importance = TRUE)
bag_pred = predict(bag_carseats, Carseats_test)
mean((Carseats_test$Sales - bag_pred)^2)
```

```
## [1] 2.953114
```

```r
importance(bag_carseats)
```

```
##               %IncMSE IncNodePurity
## CompPrice   35.238883     256.78439
## Income      10.299522     140.24737
## Advertising 23.002369     193.54415
## Population  -2.401561      69.76428
## Price       80.085452     741.31493
```

```
## ShelveLoc   80.270597     709.25579
## Age         25.943974     239.79209
## Education    2.149399      61.41957
## Urban       -1.614686      10.23359
## US           4.214299      10.17821
```

The bagging RMSE is 2.95, while the most important factors to Sales are ShelveLoc, Price, and CompPrice.

Chapter 8 - 8e

```
rf_carseats = randomForest(Sales~., data = Carseats_train, mtry = 6, ntree = 500, importance = TRUE)
rf_pred = predict(rf_carseats, Carseats_test)
mean((Carseats_test$Sales - rf_pred)^2)
```

```
## [1] 2.943268
```

```
importance(rf_carseats)
```

```
##                 %IncMSE IncNodePurity
## CompPrice   29.53839797     244.04545
## Income       7.57770678     159.01508
## Advertising 20.94657597     190.71585
## Population  -0.06395134      93.36007
## Price       69.43752343     692.73715
## ShelveLoc   64.00730181     662.61580
## Age         20.22435995     256.45854
## Education    1.73529365      73.09940
## Urban       -2.39631844      12.58273
## US           2.89048183      21.64771
```

The random forest RMSE is 2.84 for m = 6. The most important factors to Sales are ShelveLoc, Price, and CompPrice.

Chapter 8 - 11

Chapter 8 - 11a

```
rm(list = ls())
library(ISLR)

train = 1:1000
Caravan$Purchase = ifelse(Caravan$Purchase == "Yes", 1, 0)
Caravan_train = Caravan[train,]
Caravan_test = Caravan[-train,]
```

Chapter 8 - 11b

```
library(gbm)
```

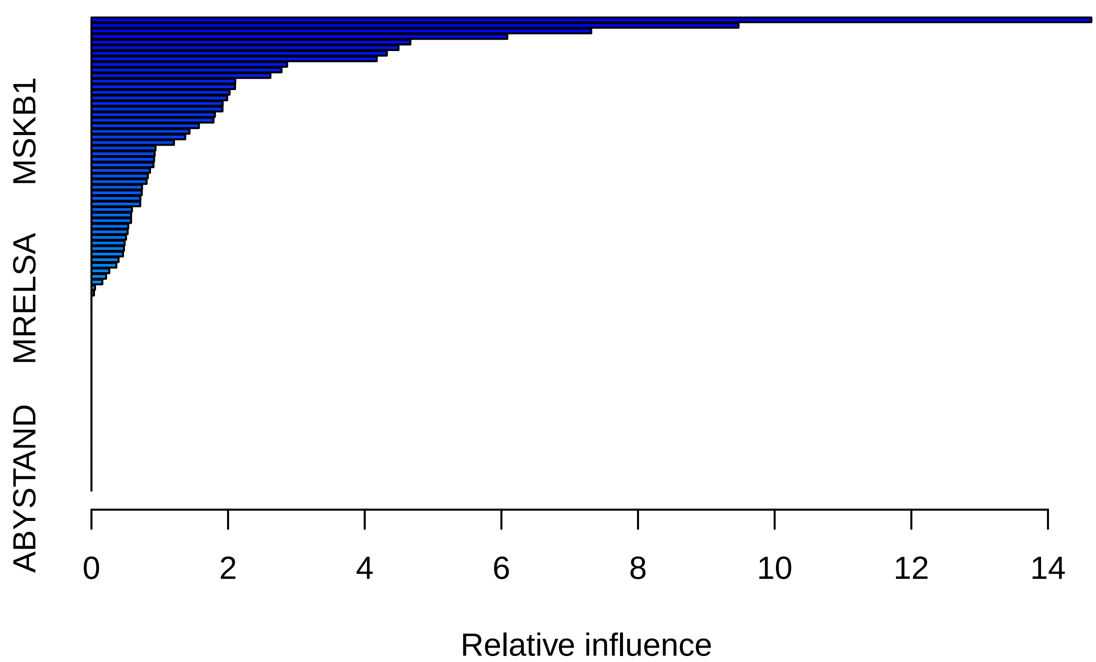
```
## Loaded gbm 2.1.8
```

```
set.seed(1)
boost.caravan = gbm(Purchase~., data = Caravan_train, n.trees = 1000, shrinkage = 0.01, distribution =
```

```
## 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.
```

```r
summary(boost.caravan)
```




```
##                 var     rel.inf
## PPERSAUT PPERSAUT 14.63504779
## MKOOPKLA MKOOPKLA  9.47091649
## MOPLHOOG MOPLHOOG  7.31457416
## MBERMIDD MBERMIDD  6.08651965
## PBRAND     PBRAND  4.66766122
## MGODGE     MGODGE  4.49463264
## ABRAND     ABRAND  4.32427755
## MINK3045 MINK3045  4.17590619
## MOSTYPE   MOSTYPE  2.86402583
## PWAPART   PWAPART  2.78191075
## MAUT1       MAUT1  2.61929152
## MBERARBG MBERARBG  2.10480508
## MSKA         MSKA  2.10185152
## MAUT2       MAUT2  2.02172510
## MSKC         MSKC  1.98684345
## MINKGEM   MINKGEM  1.92122708
## MGODPR     MGODPR  1.91777542
## MBERHOOG MBERHOOG  1.80710618
## MGODOV     MGODOV  1.78693913
## PBYSTAND PBYSTAND  1.57279593
## MSKB1       MSKB1  1.43551401
```

```
## MFWEKIND MFWEKIND  1.37264255
## MRELGE     MRELGE  1.20805179
## MOPLMIDD MOPLMIDD  0.93791970
## MINK7512 MINK7512  0.92590720
## MINK4575 MINK4575  0.91745993
## MGODRK     MGODRK  0.90765539
## MFGEKIND MFGEKIND  0.85745374
## MZPART     MZPART  0.82531066
## MRELOV     MRELOV  0.80731252
## MINKM30   MINKM30  0.74126812
## MHKOOP     MHKOOP  0.73690793
## MZFONDS   MZFONDS  0.71638323
## MAUT0       MAUT0  0.71388052
## MHHUUR     MHHUUR  0.59287247
## APERSAUT APERSAUT  0.58056986
## MOSHOOFD MOSHOOFD  0.58029563
## MSKB2       MSKB2  0.53885275
## PLEVEN     PLEVEN  0.53052444
## MINK123M MINK123M  0.50660603
## MBERARBO MBERARBO  0.48596479
## MGEMOMV   MGEMOMV  0.47614792
## PMOTSCO   PMOTSCO  0.46163590
## MSKD         MSKD  0.39735297
## MBERBOER MBERBOER  0.36417546
## MGEMLEEF MGEMLEEF  0.26166240
## MFALLEEN MFALLEEN  0.21448118
## MBERZELF MBERZELF  0.15906143
## MOPLLAAG MOPLLAAG  0.05263665
## MAANTHUI MAANTHUI  0.03766014
## MRELSA     MRELSA  0.00000000
## PWABEDR   PWABEDR  0.00000000
## PWALAND   PWALAND  0.00000000
## PBESAUT   PBESAUT  0.00000000
## PVRAAUT   PVRAAUT  0.00000000
## PAANHANG PAANHANG  0.00000000
## PTRACTOR PTRACTOR  0.00000000
## PWERKT     PWERKT  0.00000000
## PBROM       PBROM  0.00000000
## PPERSONG PPERSONG  0.00000000
## PGEZONG   PGEZONG  0.00000000
## PWAOREG   PWAOREG  0.00000000
## PZEILPL   PZEILPL  0.00000000
## PPLEZIER PPLEZIER  0.00000000
## PFIETS     PFIETS  0.00000000
## PINBOED   PINBOED  0.00000000
## AWAPART   AWAPART  0.00000000
## AWABEDR   AWABEDR  0.00000000
## AWALAND   AWALAND  0.00000000
## ABESAUT   ABESAUT  0.00000000
## AMOTSCO   AMOTSCO  0.00000000
## AVRAAUT   AVRAAUT  0.00000000
## AAANHANG AAANHANG  0.00000000
## ATRACTOR ATRACTOR  0.00000000
## AWERKT     AWERKT  0.00000000
```

```
## ABROM       ABROM  0.00000000
## ALEVEN     ALEVEN  0.00000000
## APERSONG APERSONG  0.00000000
## AGEZONG   AGEZONG  0.00000000
## AWAOREG   AWAOREG  0.00000000
## AZEILPL   AZEILPL  0.00000000
## APLEZIER APLEZIER  0.00000000
## AFIETS     AFIETS  0.00000000
## AINBOED   AINBOED  0.00000000
## ABYSTAND ABYSTAND  0.00000000
```

The most important variables are PPERSAUT, MKOOPKLA, and MOPLHOOG.

Chapter 8 - 11c

```r
boost_prob = predict(boost.caravan, Caravan_test, n.trees = 1000, type = "response")
boost_pred = ifelse(boost_prob > 0.2, 1, 0)
table(Caravan_test$Purchase, boost_pred)
```

```
##    boost_pred
##        0    1
##   0 4410  123
##   1  256   33
```

```r
33 / (33+123)
```

```
## [1] 0.2115385
```

```r
glm.caravan = glm(Purchase~., data = Caravan_train, family = binomial)
```

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

```r
glm_prob = predict(glm.caravan, Caravan_test, type = "response")
```

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

```r
glm_pred = ifelse(glm_prob > 0.2, 1, 0)
table(Caravan_test$Purchase, glm_pred)
```

```
##    glm_pred
##        0    1
##   0 4183  350
##   1  231   58
```

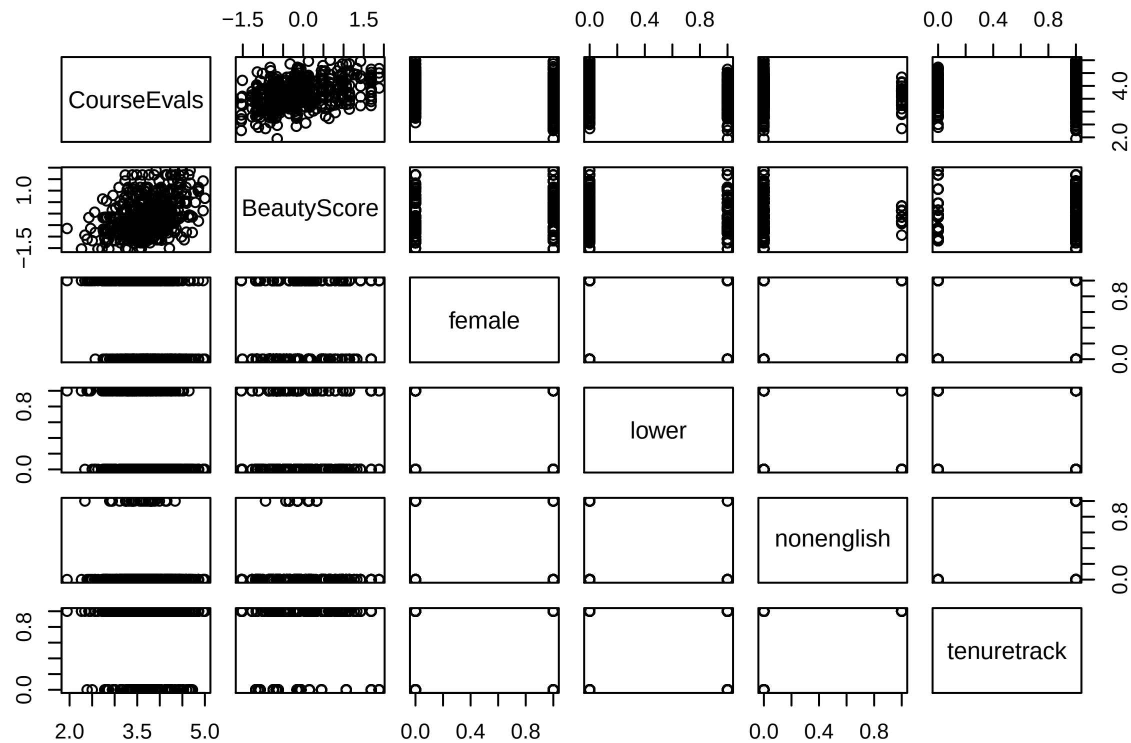
```r
58 / (350 + 58)
```

```
## [1] 0.1421569
```

Boosting predicts a correct purchase only 21.15% of the time. Logistic regression predicts a correct purchase only 14.22% of the time. This is lower than boosting prediction.

Question 1.1

```r
BeautyData <- read.csv("BeautyData.csv")

pairs(BeautyData)
```




```r
cor(BeautyData)
```

```
##              CourseEvals BeautyScore       female       lower   nonenglish
## CourseEvals   1.00000000  0.40709120 -0.231829451 -0.24864349 -0.079891096
## BeautyScore   0.40709120  1.00000000  0.125719400  0.03257686  0.010293330
## female       -0.23182945  0.12571940  1.000000000 -0.05657933  0.003805072
## lower        -0.24864349  0.03257686 -0.056579333  1.00000000 -0.143448262
## nonenglish   -0.07989110  0.01029333  0.003805072 -0.14344826  1.000000000
## tenuretrack -0.03760944 -0.01913483 -0.074315467 -0.13663972  0.134859291
##              tenuretrack
## CourseEvals -0.03760944
## BeautyScore -0.01913483
## female      -0.07431547
## lower       -0.13663972
## nonenglish   0.13485929
## tenuretrack  1.00000000
```

```r
set.seed(1)

train = sample(1:nrow(BeautyData), 0.8 * nrow(BeautyData))
BeautyTrain = BeautyData[train,]
BeautyTest = BeautyData[-train,]

lm.fit = lm(CourseEvals~., data = BeautyTrain)

lm.pred = predict(lm.fit, BeautyTest)
```

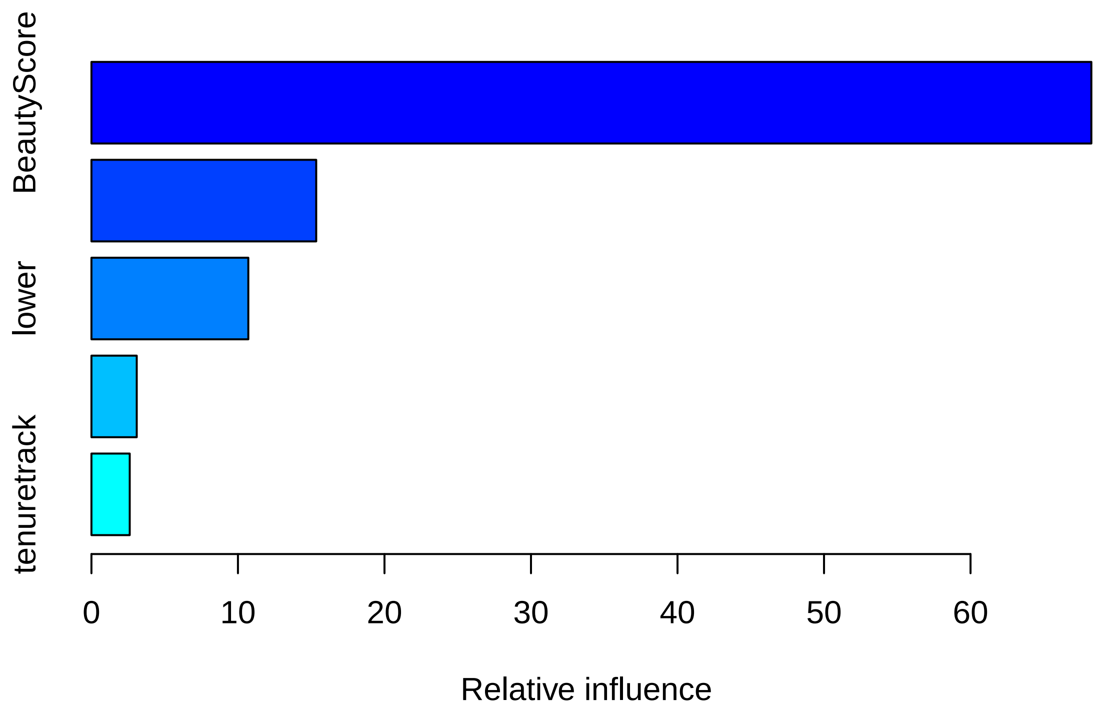
```r
sqrt(mean(lm.pred - BeautyTest$CourseEvals)^2)
```

```
## [1] 0.03064397
```

```r
summary(lm.fit)
```

```
##
## Call:
## lm(formula = CourseEvals ~ ., data = BeautyTrain)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.31494 -0.29771  0.01316  0.27795  1.06900
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.03717    0.05900  68.432  < 2e-16 ***
## BeautyScore  0.31522    0.02928  10.765  < 2e-16 ***
## female      -0.34492    0.04622  -7.463 6.30e-13 ***
## lower       -0.30300    0.04877  -6.212 1.43e-09 ***
## nonenglish  -0.30737    0.09656  -3.183  0.00158 **
## tenuretrack -0.07625    0.05529  -1.379  0.16871
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.431 on 364 degrees of freedom
## Multiple R-squared:  0.3284, Adjusted R-squared:  0.3192
## F-statistic:  35.6 on 5 and 364 DF,  p-value: < 2.2e-16
```

```r
library(tree)
library(gbm)

boost.beauty = gbm(CourseEvals~., data = BeautyData[train,], distribution = "gaussian", n.trees = 100, 

summary(boost.beauty)
```




```
##                     var   rel.inf
## BeautyScore BeautyScore 68.245732
## female           female 15.340597
## lower             lower 10.707788
## nonenglish   nonenglish  3.092445
## tenuretrack tenuretrack  2.613439
```

Beauty Score has a significant effect on course evaluation. From the boosting model, BeautyScore has a relative influence of approximately 68% on the course evaluation. However, according to the multi-linear model, almost all variables are statistically significant on the course evaluation, with the exception of tenure track. We can say that Beauty score, gender, lower division class, and native language all have an effect on course evaluation.

Question 1.2

I believe what Dr. Hamermesh is implying is that there really is no general consensus on the relationship between course evaluation and beauty. There are so many more factors that apply into something as simple as a course evaluation. Certain other factors include how well the professor teaches, the mood of the students, how students perform in the class, and many other minute details. It is unwise to simply assume that a more beautiful professor would get higher course evaluations and vice versa.

Question 2.1

```
rm(list = ls())

Housing <- read.csv("MidCity.csv")
```

```r
set.seed(1)

Housing$Brick = as.numeric(as.factor(Housing$Brick))
Housing$Nbhd = as.factor(Housing$Nbhd)


train = sample(1:nrow(Housing), 0.8 * nrow(Housing))

Housing_train = Housing[train,]
Housing_test = Housing[-train,]

lm.fit = lm(Price~.-Home, data = Housing_train)

lm.pred = predict(lm.fit, Housing_test)

summary(lm.fit)
```

```
##
## Call:
## lm(formula = Price ~ . - Home, data = Housing_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -27663.1  -7658.7   -118.2   6858.2  26828.1
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -16081.381  10368.131  -1.551  0.12425
## Nbhd2        -1399.198   2857.967  -0.490  0.62557
## Nbhd3        21732.939   3726.377   5.832 7.69e-08 ***
## Offers       -8301.346   1246.714  -6.659 1.84e-09 ***
## SqFt            54.119      6.499   8.327 6.54e-13 ***
## Brick        15560.811   2388.119   6.516 3.55e-09 ***
## Bedrooms      4200.694   1959.092   2.144  0.03460 *
## Bathrooms     8230.336   2639.627   3.118  0.00242 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10630 on 94 degrees of freedom
## Multiple R-squared:  0.8638, Adjusted R-squared:  0.8536
## F-statistic: 85.14 on 7 and 94 DF,  p-value: < 2.2e-16
```

```r
confint(lm.fit)
```

```
##                    2.5 %      97.5 %
## (Intercept) -36667.54752  4504.78500
## Nbhd2        -7073.75794  4275.36105
## Nbhd3        14334.12970 29131.74900
## Offers      -10776.72542 -5825.96694
## SqFt            41.21464    67.02252
## Brick        10819.14400 20302.47826
## Bedrooms       310.87065  8090.51799
## Bathrooms     2989.29564 13471.37663
```

According to the confidence interval, there is a 95% chance of a price premium between \$10819 and \$20302 if a house is made of brick.

Question 2.2

There does seem to be a premium for houses located in neighborhood 3 as seen in the confidence interval. There is a 95% chance of a price premium between $14334 and $29131 for houses located in neighborhood 3.

Question 2.3

```r
lm.fit2 = lm(Price~as.factor(Nbhd):Brick + Nbhd + Offers + SqFt + Brick + Bedrooms + Bathrooms, data = H

summary(lm.fit2)
```

```
##
## Call:
## lm(formula = Price ~ as.factor(Nbhd):Brick + Nbhd + Offers +
##     SqFt + Brick + Bedrooms + Bathrooms, data = Housing_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -27714.3  -5187.3   -576.1   6103.0  26166.6
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -5014.486  11317.497  -0.443  0.65875
## Nbhd2                  -7356.679   7697.528  -0.956  0.34172
## Nbhd3                  -2501.329   8550.098  -0.293  0.77053
## Offers                 -8456.577   1197.887  -7.060 3.09e-10 ***
## SqFt                      54.909      6.289   8.730 1.07e-13 ***
## Brick                   7173.604   4534.453   1.582  0.11708
## Bedrooms                4644.263   1885.790   2.463  0.01565 *
## Bathrooms               6593.208   2585.783   2.550  0.01243 *
## as.factor(Nbhd)2:Brick  5929.451   5731.604   1.035  0.30361
## as.factor(Nbhd)3:Brick 18650.198   6107.883   3.053  0.00296 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10190 on 92 degrees of freedom
## Multiple R-squared:  0.8774, Adjusted R-squared:  0.8654
## F-statistic: 73.17 on 9 and 92 DF,  p-value: < 2.2e-16
```

```r
confint(lm.fit2)
```

```
##                              2.5 %      97.5 %
## (Intercept)            -27492.01113 17463.03911
## Nbhd2                  -22644.63372  7931.27478
## Nbhd3                  -19482.56198 14479.90375
## Offers                 -10835.68384 -6077.47005
## SqFt                       42.41783    67.40037
## Brick                   -1832.21121 16179.41934
## Bedrooms                  898.92134  8389.60446
## Bathrooms                1457.62066 11728.79569
## as.factor(Nbhd)2:Brick  -5454.01029 17312.91157
## as.factor(Nbhd)3:Brick   6519.41573 30780.97961
```

According to the second confidence interval which applied a filter for brick homes in the neighborhoods, there does appear to be a price premium for brick homes in neighborhood 3. There is a 95% chance of a price premium between $6519 and $30780 for brick homes located in neighborhood 3.

Question 2.4

For the sake of prediction, it is possible that we could combine neighborhoods 1 and 2 into a single neighborhood. This might make it easier to predict prices in these neighborhoods while also eliminating some noise.

Question 3.1

There are a multitude of factors that can affect crime in a city. As seen in the Boston data set, there were 14 total factors, and we ran analyses using crime rate as a predictor. The number of police in a certain area is simply not enough to indicate crime as there are other factors to account for. Such factors include: median income of households, tax rates, youth concentration, socioeconomic conditions, and many more.

Question 3.2

UPenn researchers were able to isolate this effect by accounting for the High Alert level in D.C. They deduced that the number of crimes was not dependent on the number of police, but based off of the High Alert level and METRO ridership.

Question 3.3

The researchers used METRO ridership as a control to determine if that was affected by an increase or decrease in crime. They were also attempting to test the significance of METRO ridership on crime decrease.

Question 3.4

The researchers are attempting to estimate the decrease in crime based on High Alert in District 1, High Alert in all other districts, and METRO Ridership. It seems that High Alert in District 1 and METRO ridership are both statistically significant, but High Alert in District 1 is more significant due to it being significant at the 1% level, while METRO ridership is only significant at the 5% level.

Question 5

For the predictive modeling project, I worked with Suchit Das on implementing the univariate analysis of our prediction. We then worked on the step-wise regression together and did our best to understand the code and explain it to each other. I also contributed to the PowerPoint presentation by adding the corresponding slides and formatting it to make it aesthetically pleasing. I also contributed to the write-up report in Google Docs by adding the univariate analysis and the step-wise regressions.