Homework 2

Naga Kartheek Peddisetty, 50538422

9/27/2023

Question 1) Compare the classification performance of linear regression and k-nearest neighbor classification on the zipcode data using two digits: 0 and 8. Consider the complexity parameter “k” ranging from 1 to 17 (odd values only). Show the plotted profiles of the training and test error for each choice of k. Describe your results – are you surprised by the differences in performance?

```
rm(list = ls())

setwd("D:/Buffalo/files")

# load the training data

train_data <- data.frame(read.table(gzfile("zip.train.gz")))
dim(train_data)
```

```
## [1] 7291  257
```

```
# load the testing data

test_data <- data.frame(read.table(gzfile("zip.test.gz")))
dim(test_data)
```

```
## [1] 2007  257
```

```
sort(unique(train_data[,1]))
```

```
##  [1] 0 1 2 3 4 5 6 7 8 9
```

```
sort(unique(test_data[,1]))
```

```
##  [1] 0 1 2 3 4 5 6 7 8 9
```

```
## Taking the data of 0 and 8 in 1st column into train_data1 and test_data1.

train_data1 <- train_data[which(train_data[,1] == 0 | train_data[,1] == 8), ]
dim(train_data1)
```

```
## [1] 1736  257
```

```
test_data1 <- test_data[which(test_data[,1] == 0 | test_data[,1] == 8), ]
dim(test_data1)
```

```
## [1] 525 257
```

```
### Changing the value of 8 to 1 in all rows by using for loop in both train and test data.

n = length(train_data1[,1])
n1 = length(test_data1[,1])
for(i in 1:n){
if(train_data1[i,1] == 8)
{
  train_data1[i,1] <- 1
}
}

for(i in 1:n1){
if(test_data1[i,1] == 8)
{
  test_data1[i,1] <- 1
}
}

sort(unique(train_data1[,1]))
```

```
## [1] 0 1
```

```
sort(unique(test_data1[,1]))
```

```
## [1] 0 1
```

```
X_train <- train_data1[ ,-1]
Y_train <- train_data1[ ,1]

X_test <- test_data1[,-1]
Y_test <- test_data1[,1]

# building a classification model of linear regression.

lm.fit <- lm(V1 ~ ., data = train_data1)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = V1 ~ ., data = train_data1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.57899 -0.05232 -0.00127  0.04992  0.63509
##
## Coefficients: (1 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.8157435  4.1908947   0.910 0.362715
## V2          -0.0983247  0.1657471  -0.593 0.553124
## V3           0.0181856  0.0602984   0.302 0.763004
## V4          -0.0250276  0.0387963  -0.645 0.518961
## V5           0.0196658  0.0252856   0.778 0.436841
## V6           0.0424781  0.0170702   2.488 0.012940 *
## V7           0.0105058  0.0125129   0.840 0.401268
## V8           0.0144917  0.0102583   1.413 0.157958
## V9           0.0015242  0.0101063   0.151 0.880140
## V10         -0.0004600  0.0111881  -0.041 0.967211
## V11         -0.0060031  0.0121500  -0.494 0.621323
## V12          0.0138654  0.0139175   0.996 0.319285
## V13          0.0050452  0.0167639   0.301 0.763490
## V14          0.0079140  0.0248974   0.318 0.750633
## V15          0.0010121  0.0439101   0.023 0.981614
## V16         -0.0084990  0.0664018  -0.128 0.898172
## V17          0.3649756  0.1266995   2.881 0.004026 **
## V18          0.0819074  0.1444061   0.567 0.570663
## V19          0.1610953  0.0460359   3.499 0.000480 ***
## V20          0.0395674  0.0257641   1.536 0.124811
## V21          0.0047485  0.0167804   0.283 0.777233
## V22          0.0269528  0.0130679   2.063 0.039333 *
## V23          0.0059027  0.0107856   0.547 0.584274
## V24          0.0181518  0.0113394   1.601 0.109640
## V25          0.0109607  0.0126169   0.869 0.385133
## V26         -0.0104980  0.0117403  -0.894 0.371368
## V27          0.0080030  0.0104109   0.769 0.442185
## V28         -0.0072076  0.0108072  -0.667 0.504923
## V29         -0.0309011  0.0126976  -2.434 0.015066 *
## V30         -0.0331857  0.0176175  -1.884 0.059804 .
## V31          0.0103032  0.0293468   0.351 0.725575
## V32         -0.0195703  0.0619667  -0.316 0.752184
## V33         -0.3046893  0.1300396  -2.343 0.019259 *
## V34         -0.2109343  0.0751905  -2.805 0.005092 **
## V35          0.0217226  0.0384440   0.565 0.572129
## V36          0.0033307  0.0205231   0.162 0.871098
## V37          0.0156236  0.0147403   1.060 0.289352
## V38          0.0185258  0.0117798   1.573 0.116009
## V39          0.0012814  0.0111292   0.115 0.908352
## V40         -0.0189403  0.0108240  -1.750 0.080353 .
## V41         -0.0049995  0.0108074  -0.463 0.643717
## V42         -0.0135537  0.0103657  -1.308 0.191229
## V43         -0.0066833  0.0100414  -0.666 0.505786
## V44         -0.0206369  0.0095881  -2.152 0.031531 *
## V45         -0.0106837  0.0111232  -0.960 0.336964
```

```
## V46          -0.0084029  0.0145630  -0.577 0.564025
## V47          -0.0334236  0.0226711  -1.474 0.140619
## V48          -0.0935801  0.0438070  -2.136 0.032828 *
## V49           0.1828111  0.1074559   1.701 0.089103 .
## V50           0.1341939  0.0568574   2.360 0.018395 *
## V51           0.0216715  0.0351486   0.617 0.537615
## V52           0.0274337  0.0192782   1.423 0.154934
## V53           0.0126674  0.0146125   0.867 0.386146
## V54           0.0096549  0.0125554   0.769 0.442028
## V55           0.0357578  0.0110092   3.248 0.001188 **
## V56           0.0038971  0.0104491   0.373 0.709231
## V57          -0.0060435  0.0101143  -0.598 0.550254
## V58          -0.0062665  0.0094922  -0.660 0.509245
## V59           0.0031445  0.0090494   0.347 0.728283
## V60           0.0017757  0.0096346   0.184 0.853802
## V61           0.0081754  0.0106954   0.764 0.444758
## V62          -0.0245058  0.0127281  -1.925 0.054380 .
## V63          -0.0151359  0.0184588  -0.820 0.412357
## V64           0.0063933  0.0311919   0.205 0.837627
## V65          -0.0680100  0.1002479  -0.678 0.497612
## V66          -0.0863599  0.0524190  -1.647 0.099669 .
## V67           0.0157479  0.0278612   0.565 0.572004
## V68           0.0475468  0.0194234   2.448 0.014484 *
## V69           0.0318849  0.0148006   2.154 0.031378 *
## V70          -0.0040242  0.0129677  -0.310 0.756358
## V71           0.0044217  0.0113452   0.390 0.696783
## V72           0.0163693  0.0105657   1.549 0.121525
## V73           0.0105475  0.0100382   1.051 0.293549
## V74           0.0038191  0.0098949   0.386 0.699580
## V75          -0.0120289  0.0100127  -1.201 0.229803
## V76          -0.0081568  0.0101715  -0.802 0.422723
## V77          -0.0138299  0.0110999  -1.246 0.212981
## V78          -0.0286645  0.0125841  -2.278 0.022879 *
## V79          -0.0100144  0.0164488  -0.609 0.542734
## V80          -0.0068663  0.0246862  -0.278 0.780942
## V81          -0.0571151  0.0560385  -1.019 0.308269
## V82           0.1218355  0.0522357   2.332 0.019812 *
## V83          -0.0056045  0.0260242  -0.215 0.829519
## V84          -0.0072230  0.0201441  -0.359 0.719971
## V85           0.0178945  0.0165290   1.083 0.279157
## V86           0.0119931  0.0135638   0.884 0.376733
## V87           0.0141956  0.0117914   1.204 0.228824
## V88          -0.0149501  0.0107689  -1.388 0.165264
## V89          -0.0191684  0.0108757  -1.763 0.078190 .
## V90          -0.0034228  0.0107348  -0.319 0.749885
## V91           0.0133571  0.0110195   1.212 0.225653
## V92           0.0065063  0.0111774   0.582 0.560594
## V93          -0.0168199  0.0118337  -1.421 0.155424
## V94          -0.0106194  0.0129972  -0.817 0.414030
## V95           0.0067779  0.0167758   0.404 0.686251
## V96          -0.0170721  0.0238895  -0.715 0.474952
## V97          -0.0489725  0.0403892  -1.213 0.225509
## V98           0.0218144  0.0403169   0.541 0.588539
## V99           0.0056492  0.0269880   0.209 0.834225
## V100          0.0256198  0.0232250   1.103 0.270157
## V101          0.0165439  0.0170954   0.968 0.333334
```

```
## V102        -0.0346469  0.0141259  -2.453 0.014293 *
## V103         0.0126454  0.0119789   1.056 0.291304
## V104         0.0094921  0.0116237   0.817 0.414281
## V105         0.0345759  0.0125528   2.754 0.005951 **
## V106         0.0201607  0.0136376   1.478 0.139534
## V107        -0.0454996  0.0130305  -3.492 0.000494 ***
## V108        -0.0291044  0.0126311  -2.304 0.021350 *
## V109         0.0013036  0.0128430   0.102 0.919167
## V110        -0.0297959  0.0147966  -2.014 0.044221 *
## V111        -0.0329751  0.0177405  -1.859 0.063262 .
## V112        -0.0039970  0.0230886  -0.173 0.862584
## V113        -0.0208457  0.0331479  -0.629 0.529532
## V114        -0.0239110  0.0338435  -0.707 0.479978
## V115         0.0172440  0.0294901   0.585 0.558813
## V116        -0.0542132  0.0258019  -2.101 0.035798 *
## V117         0.0149069  0.0185811   0.802 0.422530
## V118         0.0391197  0.0145894   2.681 0.007414 **
## V119         0.0175268  0.0123016   1.425 0.154437
## V120        -0.0055773  0.0131729  -0.423 0.672072
## V121        -0.0135560  0.0144179  -0.940 0.347257
## V122         0.1514509  0.0167371   9.049  < 2e-16 ***
## V123         0.0061225  0.0142776   0.429 0.668116
## V124        -0.0050959  0.0142062  -0.359 0.719865
## V125        -0.0228183  0.0152906  -1.492 0.135832
## V126        -0.0019851  0.0177902  -0.112 0.911167
## V127         0.0337626  0.0200140   1.687 0.091823 .
## V128        -0.0490737  0.0263753  -1.861 0.063000 .
## V129        -0.0234343  0.0317108  -0.739 0.460023
## V130         0.0094441  0.0315205   0.300 0.764510
## V131         0.0143079  0.0307328   0.466 0.641600
## V132         0.0201291  0.0261226   0.771 0.441090
## V133        -0.0100093  0.0204610  -0.489 0.624778
## V134        -0.0065586  0.0161734  -0.406 0.685156
## V135         0.0030990  0.0137592   0.225 0.821830
## V136         0.0399380  0.0144147   2.771 0.005664 **
## V137         0.1145485  0.0164809   6.950 5.44e-12 ***
## V138         0.0467314  0.0165707   2.820 0.004864 **
## V139         0.0623567  0.0147212   4.236 2.42e-05 ***
## V140         0.0130099  0.0153844   0.846 0.397881
## V141         0.0114198  0.0171219   0.667 0.504897
## V142         0.0052261  0.0209510   0.249 0.803053
## V143        -0.0211565  0.0236375  -0.895 0.370911
## V144        -0.0171444  0.0284023  -0.604 0.546185
## V145        -0.0212524  0.0320623  -0.663 0.507531
## V146        -0.0448038  0.0308468  -1.452 0.146585
## V147        -0.0138053  0.0313685  -0.440 0.659927
## V148        -0.0449893  0.0266706  -1.687 0.091843 .
## V149        -0.0171961  0.0212496  -0.809 0.418507
## V150        -0.0153721  0.0167919  -0.915 0.360107
## V151         0.0192319  0.0151760   1.267 0.205262
## V152        -0.0272317  0.0160441  -1.697 0.089849 .
## V153         0.0664562  0.0161875   4.105 4.26e-05 ***
## V154         0.0393235  0.0148524   2.648 0.008192 **
## V155         0.0380471  0.0141334   2.692 0.007182 **
## V156        -0.0097080  0.0152531  -0.636 0.524573
## V157        -0.0153869  0.0171900  -0.895 0.370877
```

```
## V158        -0.0081585  0.0210594  -0.387 0.698514
## V159         0.0223716  0.0232620   0.962 0.336347
## V160         0.0199742  0.0295907   0.675 0.499770
## V161         0.0311880  0.0300919   1.036 0.300173
## V162        -0.0214908  0.0321316  -0.669 0.503703
## V163        -0.0261142  0.0309263  -0.844 0.398581
## V164         0.0053485  0.0250796   0.213 0.831154
## V165         0.0010737  0.0199901   0.054 0.957172
## V166        -0.0227576  0.0166507  -1.367 0.171907
## V167        -0.0536039  0.0157095  -3.412 0.000662 ***
## V168        -0.0143710  0.0155249  -0.926 0.354765
## V169         0.0060829  0.0148153   0.411 0.681441
## V170         0.0258426  0.0144107   1.793 0.073130 .
## V171        -0.0219717  0.0135692  -1.619 0.105610
## V172         0.0059285  0.0142113   0.417 0.676615
## V173         0.0046369  0.0153890   0.301 0.763220
## V174         0.0268739  0.0191832   1.401 0.161450
## V175         0.0130561  0.0207778   0.628 0.529860
## V176        -0.0028478  0.0274733  -0.104 0.917456
## V177         0.0440941  0.0307963   1.432 0.152413
## V178        -0.0185685  0.0392220  -0.473 0.635983
## V179         0.0167798  0.0276508   0.607 0.544046
## V180         0.0120754  0.0224368   0.538 0.590522
## V181        -0.0027394  0.0177402  -0.154 0.877302
## V182        -0.0120590  0.0150270  -0.802 0.422398
## V183         0.0114369  0.0140877   0.812 0.417016
## V184        -0.0341679  0.0147252  -2.320 0.020456 *
## V185         0.0129030  0.0147777   0.873 0.382727
## V186         0.0155142  0.0133357   1.163 0.244870
## V187        -0.0120870  0.0122114  -0.990 0.322427
## V188        -0.0028380  0.0120839  -0.235 0.814351
## V189        -0.0113206  0.0140086  -0.808 0.419153
## V190        -0.0286606  0.0176261  -1.626 0.104157
## V191        -0.0133148  0.0194453  -0.685 0.493620
## V192         0.0184563  0.0239891   0.769 0.441801
## V193         0.0123853  0.0364927   0.339 0.734364
## V194         0.0060315  0.0636749   0.095 0.924548
## V195        -0.0075929  0.0275464  -0.276 0.782862
## V196        -0.0559349  0.0215795  -2.592 0.009635 **
## V197        -0.0448723  0.0174630  -2.570 0.010280 *
## V198        -0.0127179  0.0136047  -0.935 0.350036
## V199         0.0093573  0.0117860   0.794 0.427362
## V200        -0.0181812  0.0133135  -1.366 0.172265
## V201         0.0308698  0.0144182   2.141 0.032435 *
## V202        -0.0101061  0.0121712  -0.830 0.406485
## V203         0.0166536  0.0111993   1.487 0.137223
## V204         0.0165876  0.0119010   1.394 0.163588
## V205        -0.0005716  0.0133134  -0.043 0.965763
## V206         0.0255817  0.0162161   1.578 0.114883
## V207         0.0169977  0.0186352   0.912 0.361848
## V208        -0.0145957  0.0253022  -0.577 0.564125
## V209        -0.0814827  0.0602223  -1.353 0.176252
## V210        -0.2406848  0.2253841  -1.068 0.285746
## V211        -0.0582537  0.0422268  -1.380 0.167936
## V212        -0.0080222  0.0237720  -0.337 0.735816
## V213         0.0143300  0.0186484   0.768 0.442354
```

```
## V214         0.0169596  0.0138481   1.225 0.220887
## V215        -0.0027657  0.0118883  -0.233 0.816072
## V216         0.0017041  0.0122237   0.139 0.889147
## V217         0.0023148  0.0136403   0.170 0.865268
## V218        -0.0047178  0.0133766  -0.353 0.724373
## V219        -0.0043898  0.0124999  -0.351 0.725499
## V220         0.0101299  0.0132255   0.766 0.443835
## V221         0.0175364  0.0142270   1.233 0.217915
## V222         0.0122844  0.0159070   0.772 0.440081
## V223         0.0328380  0.0210228   1.562 0.118497
## V224         0.0673227  0.0367065   1.834 0.066843 .
## V225         0.2924740  0.1820179   1.607 0.108302
## V226        -0.0465234  0.7045725  -0.066 0.947362
## V227         0.1550182  0.1205702   1.286 0.198746
## V228        -0.0083267  0.0405339  -0.205 0.837269
## V229        -0.0387880  0.0236319  -1.641 0.100939
## V230        -0.0193378  0.0179248  -1.079 0.280840
## V231        -0.0230144  0.0154388  -1.491 0.136257
## V232         0.0191096  0.0158534   1.205 0.228243
## V233        -0.0434135  0.0182549  -2.378 0.017524 *
## V234        -0.0146745  0.0194300  -0.755 0.450221
## V235        -0.0202198  0.0165164  -1.224 0.221062
## V236         0.0064315  0.0140542   0.458 0.647292
## V237         0.0302232  0.0146379   2.065 0.039123 *
## V238         0.0091060  0.0198333   0.459 0.646210
## V239        -0.0221494  0.0369824  -0.599 0.549320
## V240        -0.0871473  0.1156912  -0.753 0.451405
## V241        -0.2018107  0.1611164  -1.253 0.210557
## V242         0.1683008  3.9598868   0.043 0.966105
## V243        -0.1763183  1.1246913  -0.157 0.875447
## V244        -0.1864563  0.0957196  -1.948 0.051611 .
## V245        -0.0103017  0.0466946  -0.221 0.825419
## V246        -0.0085338  0.0283880  -0.301 0.763753
## V247         0.0256661  0.0200434   1.281 0.200561
## V248        -0.0237846  0.0174466  -1.363 0.173000
## V249        -0.0179776  0.0169474  -1.061 0.288960
## V250        -0.0083003  0.0164502  -0.505 0.613935
## V251         0.0294613  0.0153274   1.922 0.054780 .
## V252        -0.0084298  0.0166814  -0.505 0.613393
## V253         0.0182203  0.0227422   0.801 0.423163
## V254         0.0357196  0.0434482   0.822 0.411142
## V255         0.0586233  0.1053780   0.556 0.578080
## V256         3.4021304  2.2378259   1.520 0.128653
## V257                NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1174 on 1480 degrees of freedom
## Multiple R-squared:  0.9453, Adjusted R-squared:  0.9359
## F-statistic: 100.4 on 255 and 1480 DF,  p-value: < 2.2e-16
```

```
y_pred_test <- predict(lm.fit, newdata = X_test)
```

```
## Warning in predict.lm(lm.fit, newdata = X_test): prediction from rank-deficient
## fit; attr(*, "non-estim") has doubtful cases
```

```
y_pred_test <- ifelse(y_pred_test > 0.5, yes = 1, no = 0)
table(Y_test,y_pred_test)
```

```
##       y_pred_test
## Y_test   0   1
##      0 350   9
##      1   6 160
```

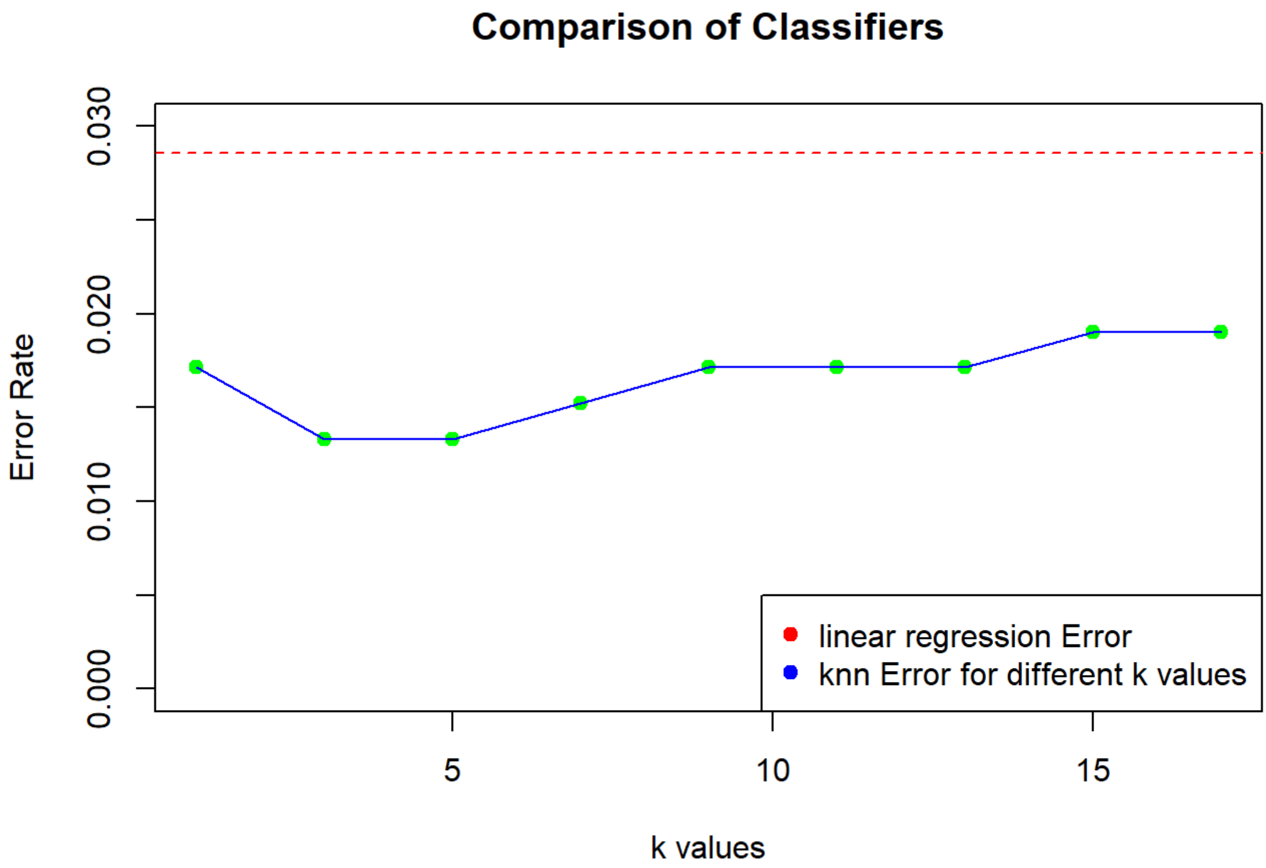
```
error_lm <- mean(y_pred_test != Y_test)
error_lm
```

```
## [1] 0.02857143
```

```
# building a classification model using knn

library(class)
k <- seq(from = 1, to = 17, by = 2)
k_error <- rep(NA, length(k))
for (i in 1:length(k)) {
    y_pred_test_knn <- knn(X_train, X_test, Y_train, k[i])
    k_error[i] <- mean(y_pred_test_knn != Y_test)
}

## Creating error matrix for linear regression and knn model with different k values.

error_mat <- matrix(c(error_lm, k_error), ncol = 1)
colnames(error_mat) <- c("Error Rate")
rownames(error_mat) <- c("Linear Regression", paste("k-NN with k =", k))
error_mat
```

```
##                   Error Rate
## Linear Regression 0.02857143
## k-NN with k = 1   0.01714286
## k-NN with k = 3   0.01333333
## k-NN with k = 5   0.01333333
## k-NN with k = 7   0.01523810
## k-NN with k = 9   0.01714286
## k-NN with k = 11  0.01714286
## k-NN with k = 13  0.01714286
## k-NN with k = 15  0.01904762
## k-NN with k = 17  0.01904762
```

```
### PLot for the comparison of error rates

plot(c(1, 17), c(0,0.03), type = "n", main = "Comparison of Classifiers",
     ylab = "Error Rate", xlab = "k values",pch = 19)
abline(h = error_lm, col = "red", lty = 2)
points(k, k_error, col = "green", pch = 19)
lines(k, k_error, col = "blue", lty = 1)
legend("bottomright", legend = c("linear regression Error", "knn Error for different k value
s"), col = c("red", "blue"), pch = 19)
```



Compared to the linear regression model, test error is less in knn model for the different values of k.

From the above plot, we can see that knn model performed well for k values k = 3,5 with error of 0.0133 and linear regression with error of 0.0285

When we are using knn classifier, it is better to use low k values.

Question 2) This question should be answered using the Carseats data set.

a) Fit a multiple regression model to predict Sales using Price, Urban, and US.

```
library(ISLR2)
data(Carseats)
dim(Carseats)  ### 400 * 11
```

```
## [1] 400  11
```

```r
str(Carseats)
```

```
## 'data.frame':    400 obs. of  11 variables:
##  $ Sales      : num  9.5 11.22 10.06 7.4 4.15 ...
##  $ CompPrice  : num  138 111 113 117 141 124 115 136 132 132 ...
##  $ Income     : num  73 48 35 100 64 113 105 81 110 113 ...
##  $ Advertising: num  11 16 10 4 3 13 0 15 0 0 ...
##  $ Population : num  276 260 269 466 340 501 45 425 108 131 ...
##  $ Price      : num  120 83 80 97 128 72 108 120 124 124 ...
##  $ ShelveLoc  : Factor w/ 3 levels "Bad","Good","Medium": 1 2 3 3 1 1 3 2 3 3 ...
##  $ Age        : num  42 65 59 55 38 78 71 67 76 76 ...
##  $ Education  : num  17 10 12 14 13 16 15 10 10 17 ...
##  $ Urban      : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 2 2 1 1 ...
##  $ US         : Factor w/ 2 levels "No","Yes": 2 2 2 2 1 2 1 2 1 2 ...
```

```r
head(Carseats)
```

```
##   Sales CompPrice Income Advertising Population Price ShelveLoc Age Education
## 1  9.50       138     73          11        276   120       Bad  42        17
## 2 11.22       111     48          16        260    83      Good  65        10
## 3 10.06       113     35          10        269    80    Medium  59        12
## 4  7.40       117    100           4        466    97    Medium  55        14
## 5  4.15       141     64           3        340   128       Bad  38        13
## 6 10.81       124    113          13        501    72       Bad  78        16
##   Urban  US
## 1   Yes Yes
## 2   Yes Yes
## 3   Yes Yes
## 4   Yes Yes
## 5   Yes  No
## 6    No Yes
```

```r
lm.fit <- lm(Sales ~ Price + Urban + US, data = Carseats)

contrasts(Carseats$Urban)
```

```
##     Yes
## No    0
## Yes   1
```

```r
contrasts(Carseats$US)
```

```
##     Yes
## No    0
## Yes   1
```

b) Provide an interpretation of each coefficient in the model. Be careful—some of the variables in the model are qualitative!

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

```
##
## Call:
## lm(formula = Sales ~ Price + Urban + US, data = Carseats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9206 -1.6220 -0.0564  1.5786  7.0581
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.043469   0.651012  20.036  < 2e-16 ***
## Price       -0.054459   0.005242 -10.389  < 2e-16 ***
## UrbanYes    -0.021916   0.271650  -0.081    0.936
## USYes        1.200573   0.259042   4.635 4.86e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.472 on 396 degrees of freedom
## Multiple R-squared:  0.2393, Adjusted R-squared:  0.2335
## F-statistic: 41.52 on 3 and 396 DF,  p-value: < 2.2e-16
```

Price : The linear regression suggests a relationship between sales and price given the low p-value of t statistic. The coefficient shows the negative sign means negative relationship between price and sales: if price increases, sales decreases. The effect of a 1-unit increase in Price is a change in Sales of -0.054 units (54 sales).

Urban : The linear regression suggests there isn’t a relationship between sales and urban given the p value of t statistic for Urban is 0.936 almost close to 1 and the effect of a store being in an urban area is a change in Sales of -0.0219 units (21.9 sales).

US : The linear regression suggests a relationship between amount of sales and the store is in US given the low p-value of t statistic.The coefficient shows the positive sign means positive relationship between US and sales: if the store is in the US, the sales will increase by approximately 1200 units.

c) Write out the model in equation form, being careful to handle the qualitative variables properly.

Here both Urban and US are qualitative(Factors) variables.

$$Urban = \begin{cases} 0 & ifNo \\ 1 & ifYes \end{cases}$$

$$US = \begin{cases} 0 & ifNo \\ 1 & ifYes \end{cases}$$

```
### Sales = β0 + β1 * Price + β2 * Urban + β3 * US + ϵ

### Sales = 13.043469 - 0.054459 * Price - 0.021916 * UrbanYes + 1.200573 * USYes + ϵ
```

d) For which of the predictors can you reject the null hypothesis H0 : βj = 0?

```
## For the predictors Price and US we can reject the null hypothesis based on the low p value of t statistic.
```

e) On the basis of your response to the previous question, fit a smaller model that only uses the predictors for which there is evidence of association with the outcome.

```
lm.smallerfit <- lm(Sales ~ Price + US, data = Carseats)
summary(lm.smallerfit)
```

```
##
## Call:
## lm(formula = Sales ~ Price + US, data = Carseats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9269 -1.6286 -0.0574  1.5766  7.0515
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.03079    0.63098  20.652  < 2e-16 ***
## Price       -0.05448    0.00523 -10.416  < 2e-16 ***
## USYes        1.19964    0.25846   4.641 4.71e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.469 on 397 degrees of freedom
## Multiple R-squared:  0.2393, Adjusted R-squared:  0.2354
## F-statistic: 62.43 on 2 and 397 DF,  p-value: < 2.2e-16
```

f) How well do the models in (a) and (e) fit the data?

```
# For Part a) :
### RSE = 2.472 and Adjusted R square = 0.2335

# For Part e) :
### RSE = 2.469 and Adjusted R square = 0.2354

# Based on the RSE and R^2 of the linear regressions, they both fit the data similarly, But w
ith linear regression from (e: RSE reduced and Adjusted R square increased) fitting the data
slightly better.
```

g) Using the model from (e), obtain 95 % confidence intervals for the coefficient(s).

```
confint(lm.smallerfit)
```

```
##                   2.5 %      97.5 %
## (Intercept) 11.79032020 14.27126531
## Price       -0.06475984 -0.04419543
## USYes        0.69151957  1.70776632
```

We can say, that there is a 95% probability that the true parameter for Price (β1) falls within the interval: (-0.0647, -0.0441) and a 5% probability that it doesn't.

Question 3) This problem involves the Boston data set, which we saw in the lab for this chapter. We will now try to predict per capita crime rate using the other variables in this data set. In other words, per capita crime rate is the response, and the other variables are the predictors.

a) For each predictor, fit a simple linear regression model to predict the response. Describe your results. In which of the models is there a statistically significant association between the predictor and the response? Create some plots to back up your assertions.

```
library(MASS)
```

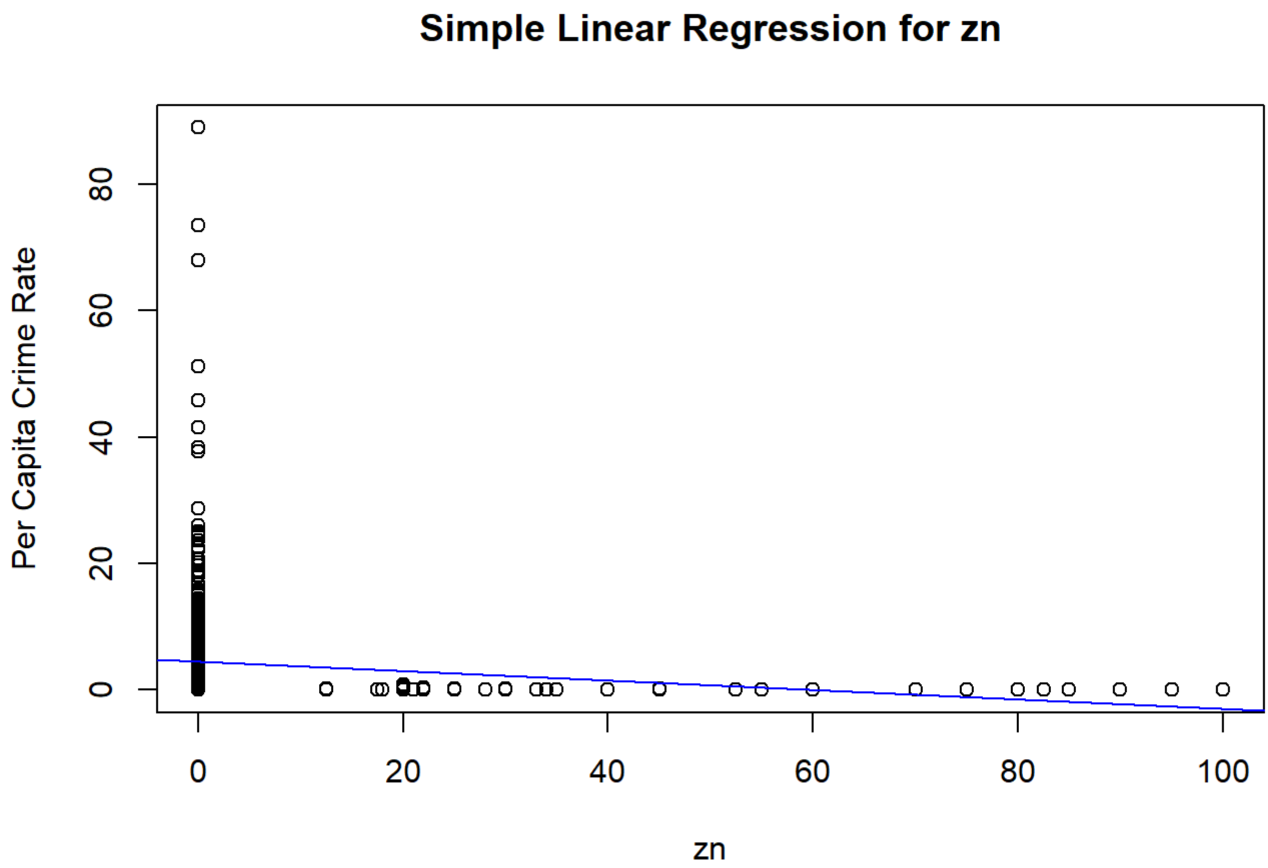
```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:ISLR2':
##
##     Boston
```

```
data("Boston")
dim(Boston)      ## 506 * 14
```

```
## [1] 506  14
```

```
## str(Boston)
## summary(Boston)

lm.fit1 <- lm(crim ~ zn, data = Boston)

plot(Boston$zn , Boston$crim, xlab = "zn", ylab = "Per Capita Crime Rate", main = "Simple Lin
ear Regression for zn")
abline(lm.fit1, col = "blue")
```

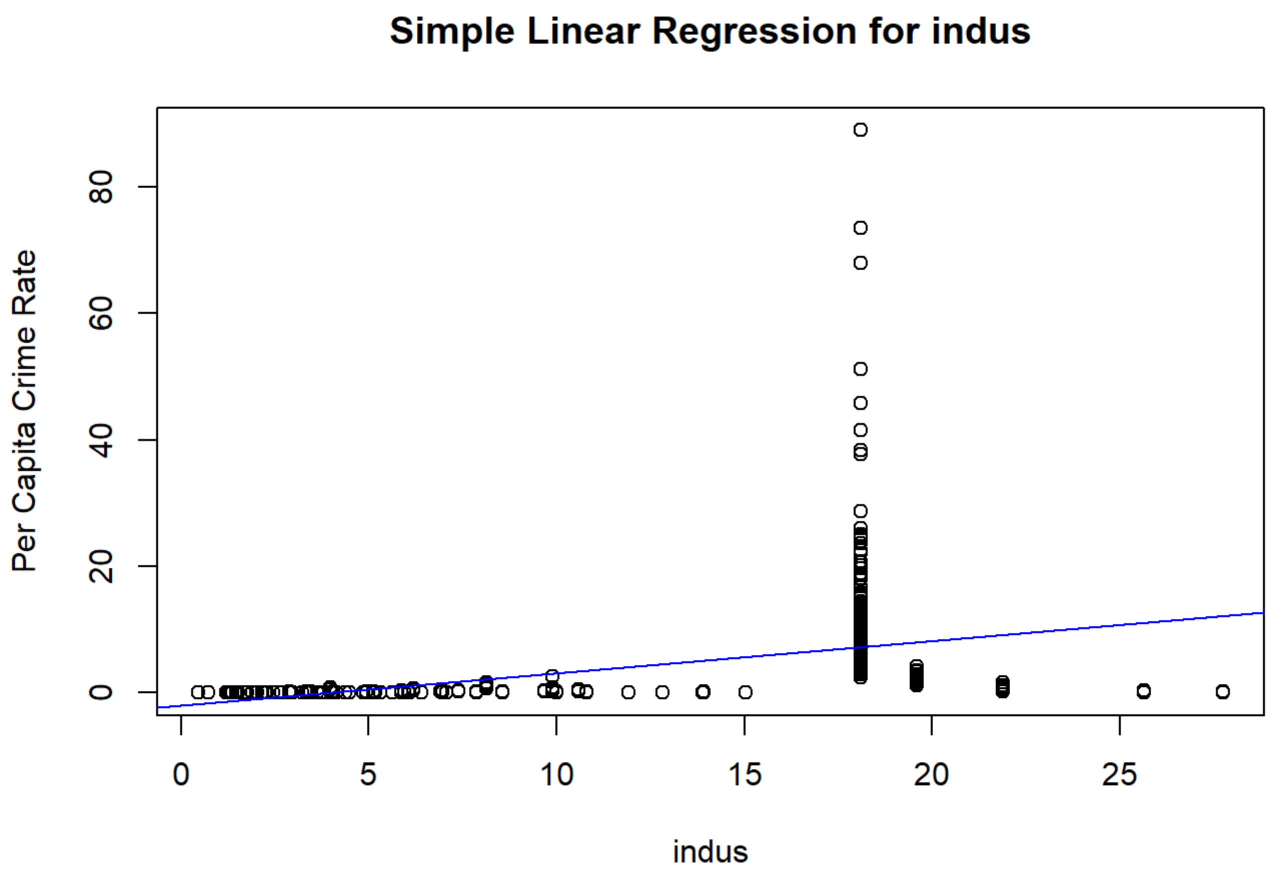


```
summary(lm.fit1)
```

```
##
## 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
```

According to the summary the p-value is low and regression coefficient is -0.07393. Means there is a negative relationship between zn and crim. Clearly, we can see that in the above plot.

```
lm.fit2 <- lm(crim ~ indus, data = Boston)

plot(Boston$indus , Boston$crim, xlab = "indus", ylab = "Per Capita Crime Rate", main = "Simp
le Linear Regression for indus")
abline(lm.fit2, col = "blue")
```

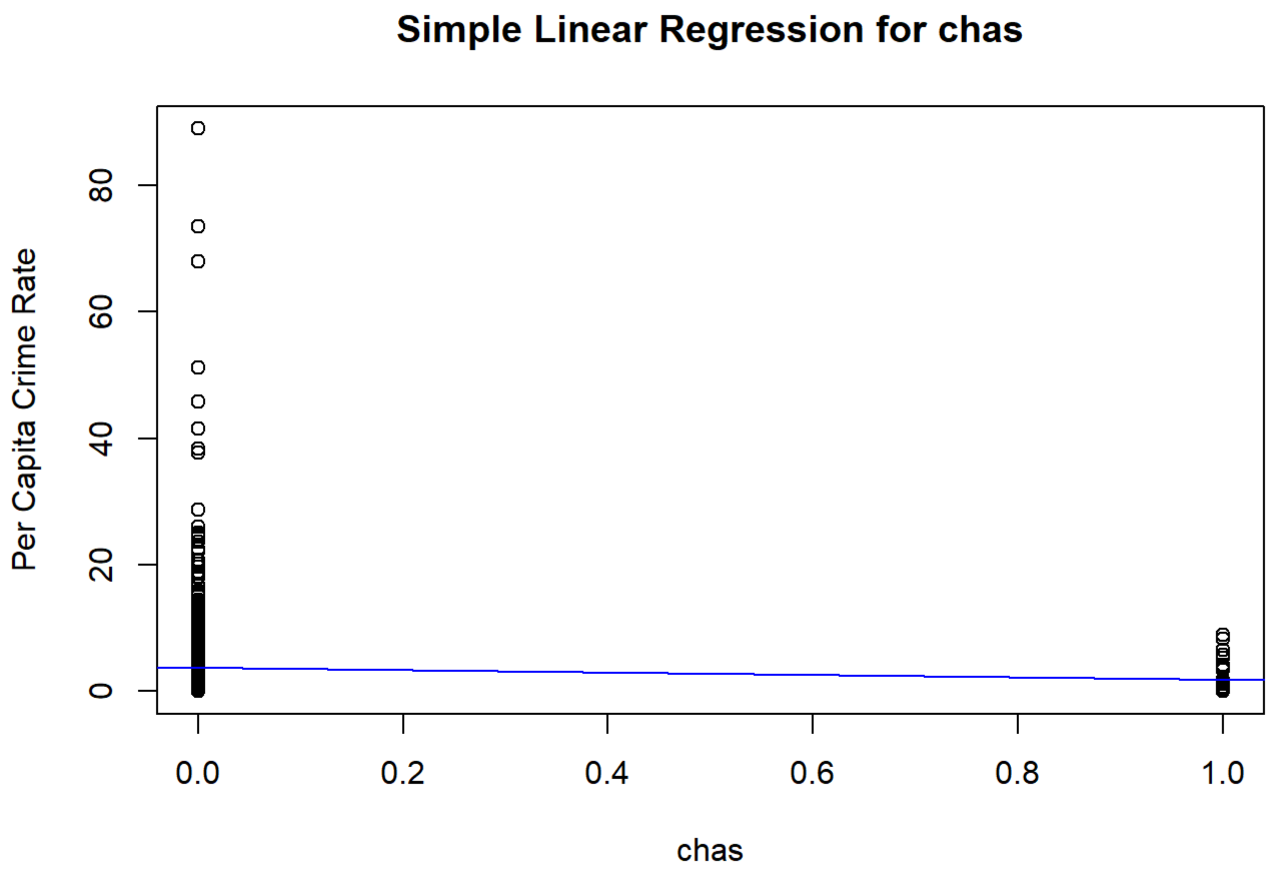



```
summary(lm.fit2)
```

```
##
## 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
```

According to the summary the p-value is low and regression coefficient is 0.50978. Means there is a positive relationship between indus and crim. Clearly, we can see that in the above plot.

```
lm.fit3 <- lm(crim ~ chas, data = Boston)

plot(Boston$chas , Boston$crim, xlab = "chas", ylab = "Per Capita Crime Rate", main = "Simple Linear Regression for chas")
abline(lm.fit3, col = "blue")
```

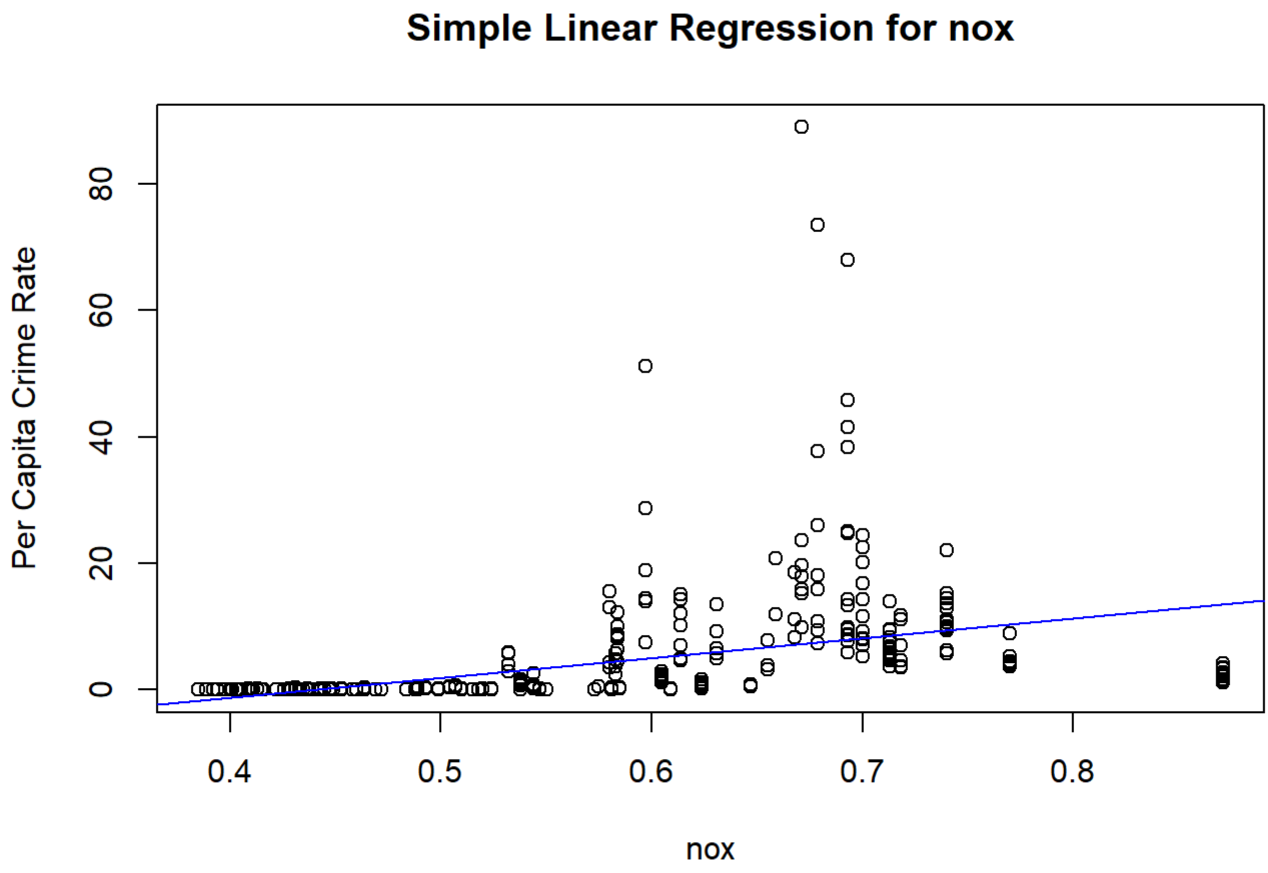



```
summary(lm.fit3)
```

```
##
## 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
```

According to the summary the p-value is 0.209 which is least significant. Means there isn't a relationship between chas and crim. Clearly, we can see that in the above plot.

```
lm.fit4 <- lm(crim ~ nox, data = Boston)

plot(Boston$nox , Boston$crim, xlab = "nox", ylab = "Per Capita Crime Rate", main = "Simple L
inear Regression for nox")
abline(lm.fit4, col = "blue")
```

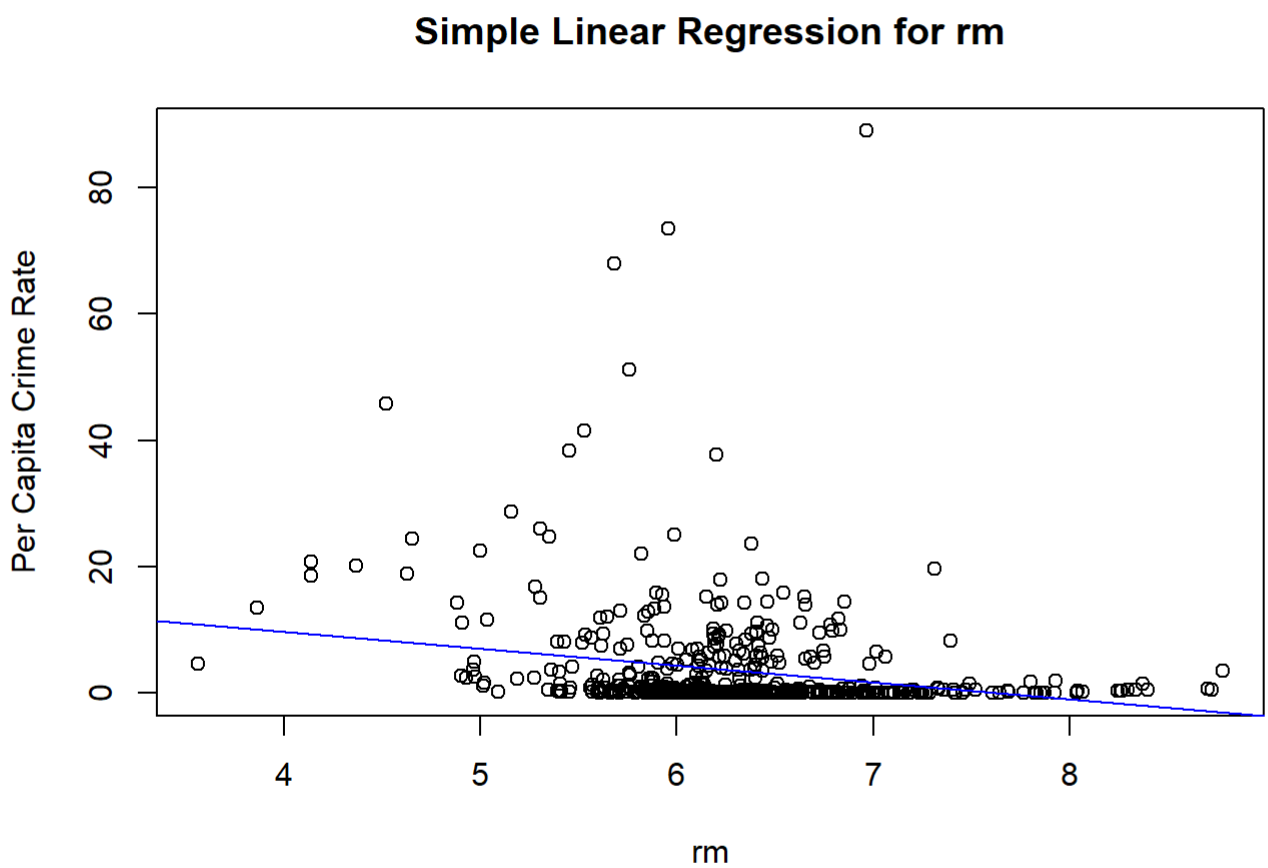



```
summary(lm.fit4)
```

```
##
## 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
```

According to the summary the p-value is low and regression coefficient is 31.249. Means there is a positive relationship between nox and crim. Clearly, we can see that in the above plot.

```
lm.fit5 <- lm(crim ~ rm, data = Boston)

plot(Boston$rm , Boston$crim, xlab = "rm", ylab = "Per Capita Crime Rate", main = "Simple Lin
ear Regression for rm")
abline(lm.fit5, col = "blue")
```

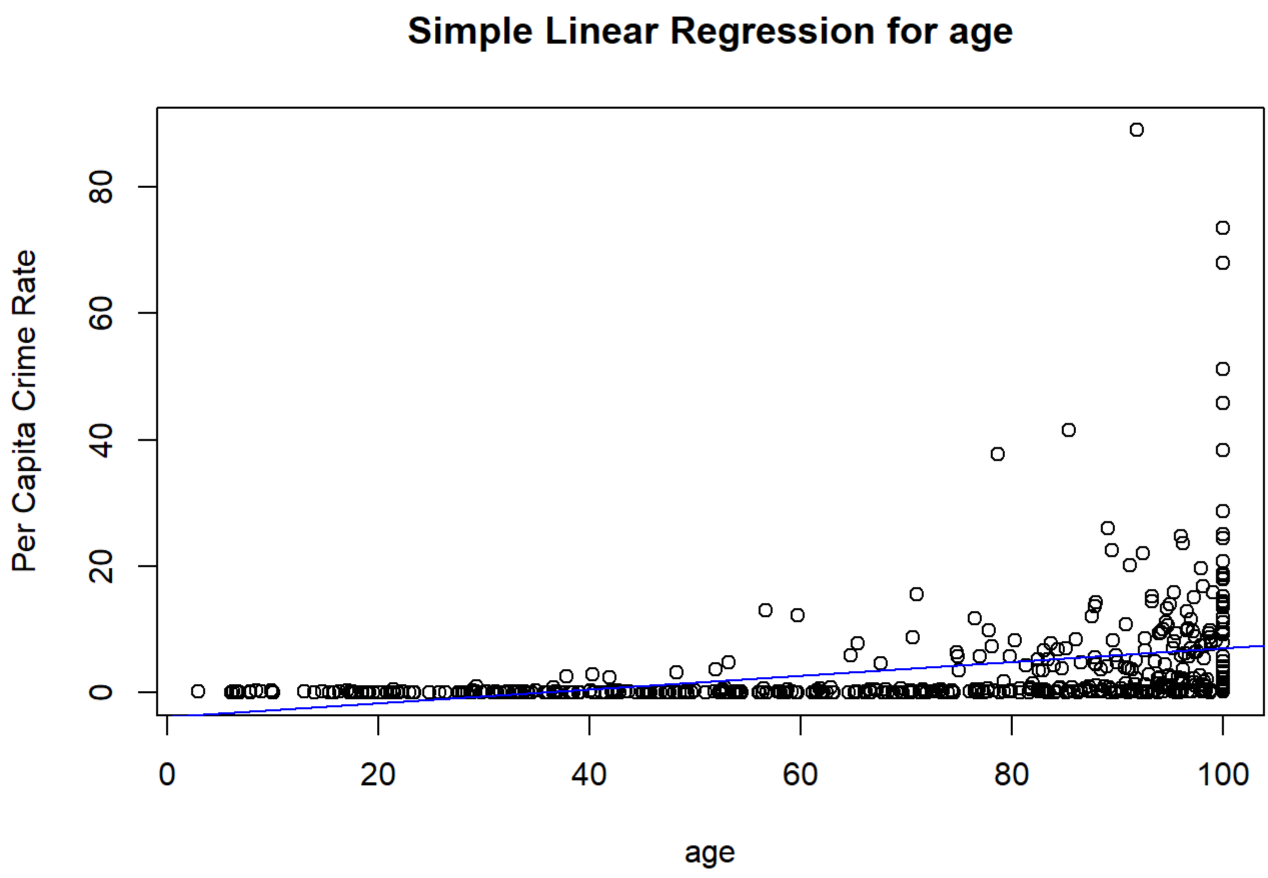


```
summary(lm.fit5)
```

```
##
## 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
```

According to the summary the p-value is low and regression coefficient is -2.684. Means there is a negative relationship between rm and crim. Clearly, we can see that in the above plot.

```
lm.fit6 <- lm(crim ~ age, data = Boston)

plot(Boston$age , Boston$crim, xlab = "age", ylab = "Per Capita Crime Rate", main = "Simple L
inear Regression for age")
abline(lm.fit6, col = "blue")
```

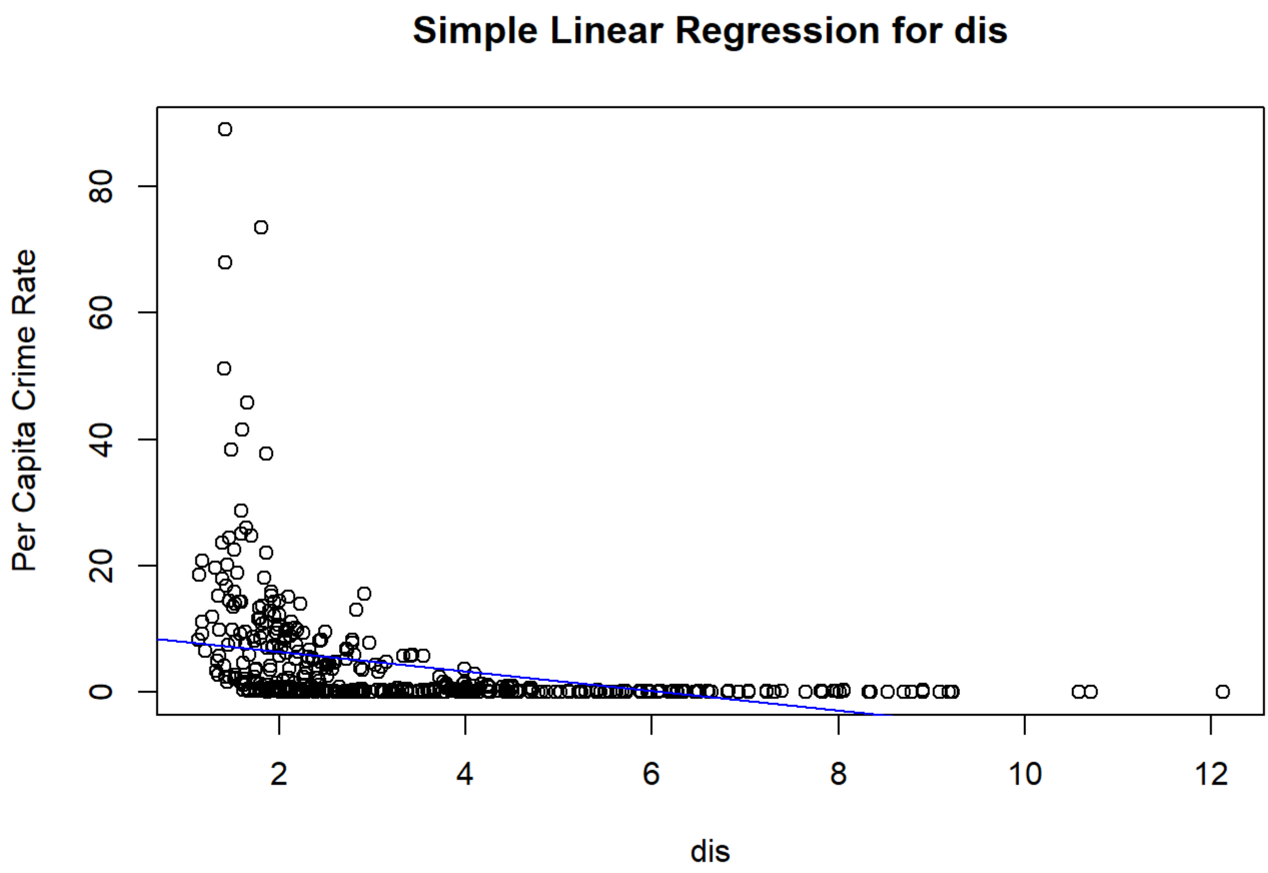



```
summary(lm.fit6)
```

```
##
## 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
```

According to the summary the p-value is low and regression coefficient is 0.10779. Means there is a positive relationship between age and crim. Clearly, we can see that in the above plot.

```
lm.fit7 <- lm(crim ~ dis, data = Boston)

plot(Boston$dis , Boston$crim, xlab = "dis", ylab = "Per Capita Crime Rate", main = "Simple L
inear Regression for dis")
abline(lm.fit7, col = "blue")
```

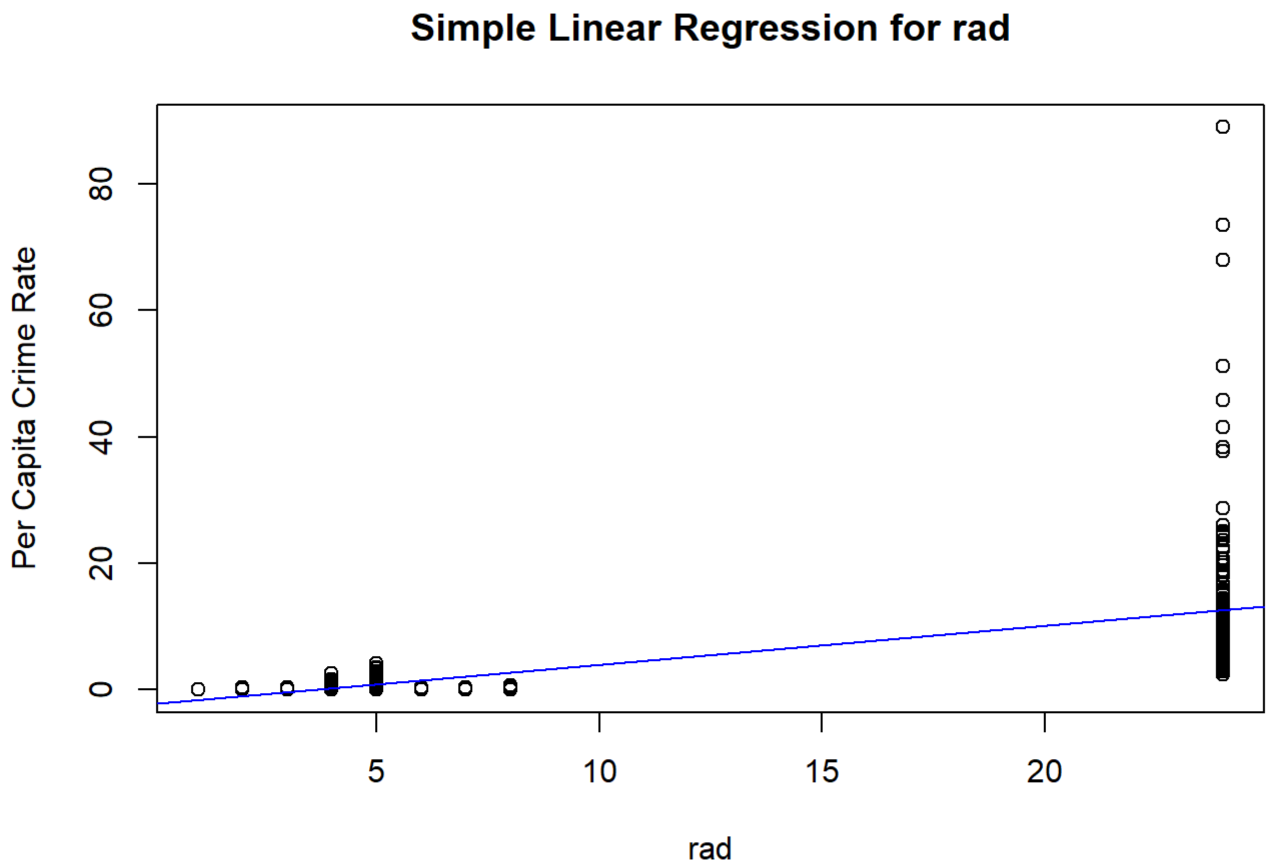



```
summary(lm.fit7)
```

```
##
## 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
```

According to the summary the p-value is low and regression coefficient is -1.5509. Means there is a negative relationship between dis and crim. Clearly, we can see that in the above plot.

```
lm.fit8 <- lm(crim ~ rad, data = Boston)

plot(Boston$rad , Boston$crim, xlab = "rad", ylab = "Per Capita Crime Rate", main = "Simple L
inear Regression for rad")
abline(lm.fit8, col = "blue")
```

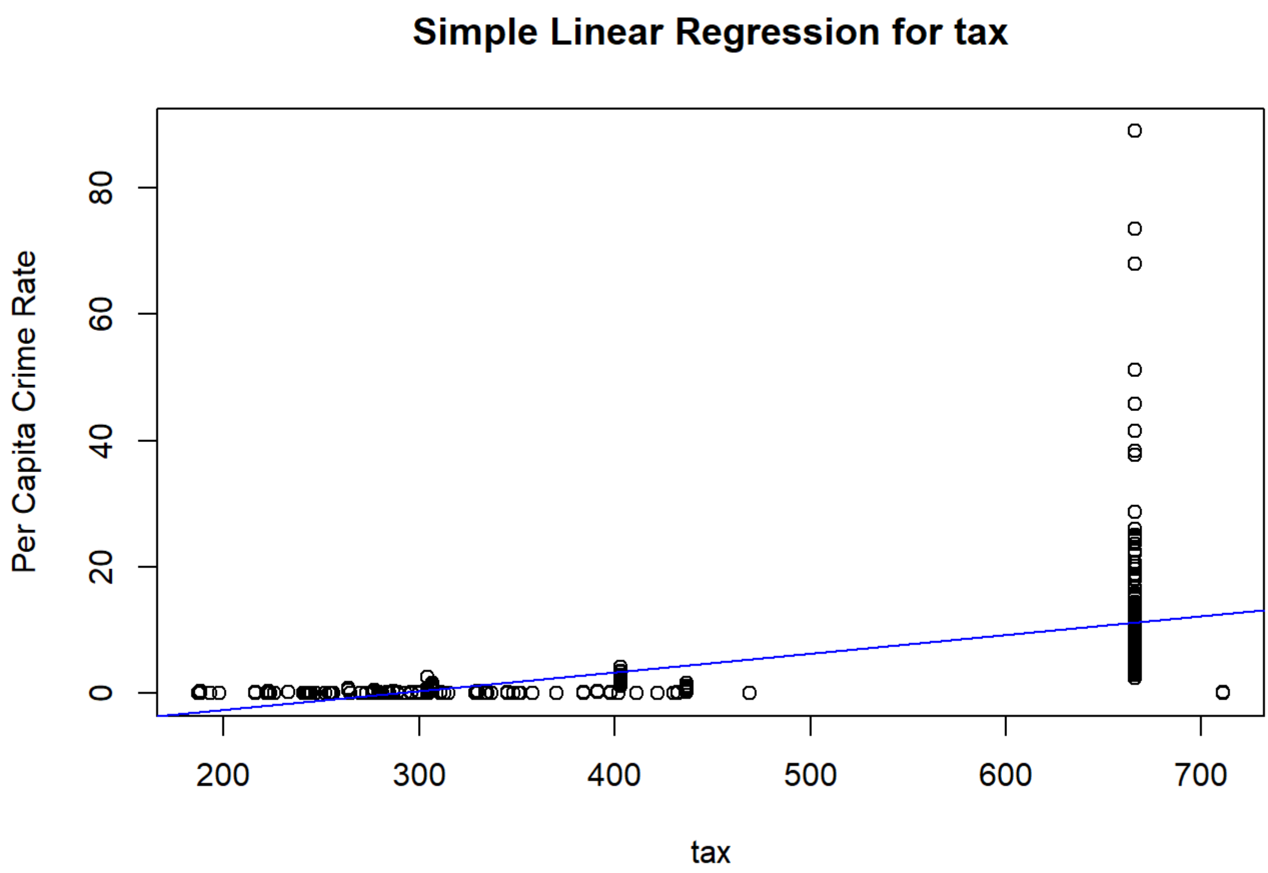



```
summary(lm.fit8)
```

```
##
## 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
```

According to the summary the p-value is low and regression coefficient is 0.61791. Means there is a positive relationship between rad and crim. Clearly, we can see that in the above plot.

```
lm.fit9 <- lm(crim ~ tax, data = Boston)

plot(Boston$tax , Boston$crim, xlab = "tax", ylab = "Per Capita Crime Rate", main = "Simple L
inear Regression for tax")
abline(lm.fit9, col = "blue")
```

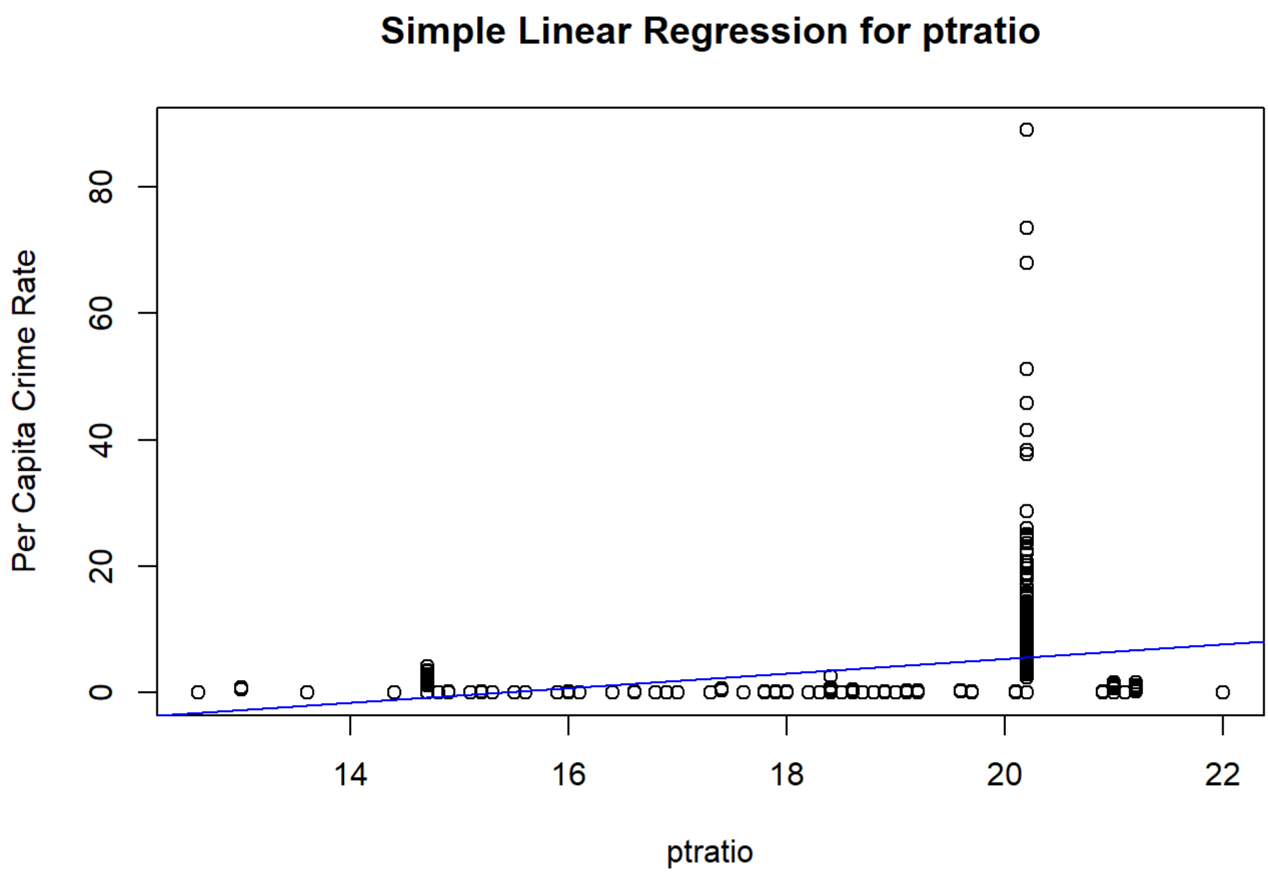


```
summary(lm.fit9)
```

```
##
## 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
```

According to the summary the p-value is low and regression coefficient is 0.029742. Means there is a positive relationship between tax and crim. Clearly, we can see that in the above plot.

```
lm.fit10 <- lm(crim ~ ptratio, data = Boston)

plot(Boston$ptratio , Boston$crim, xlab = "ptratio", ylab = "Per Capita Crime Rate", main =
"Simple Linear Regression for ptratio")
abline(lm.fit10, col = "blue")
```

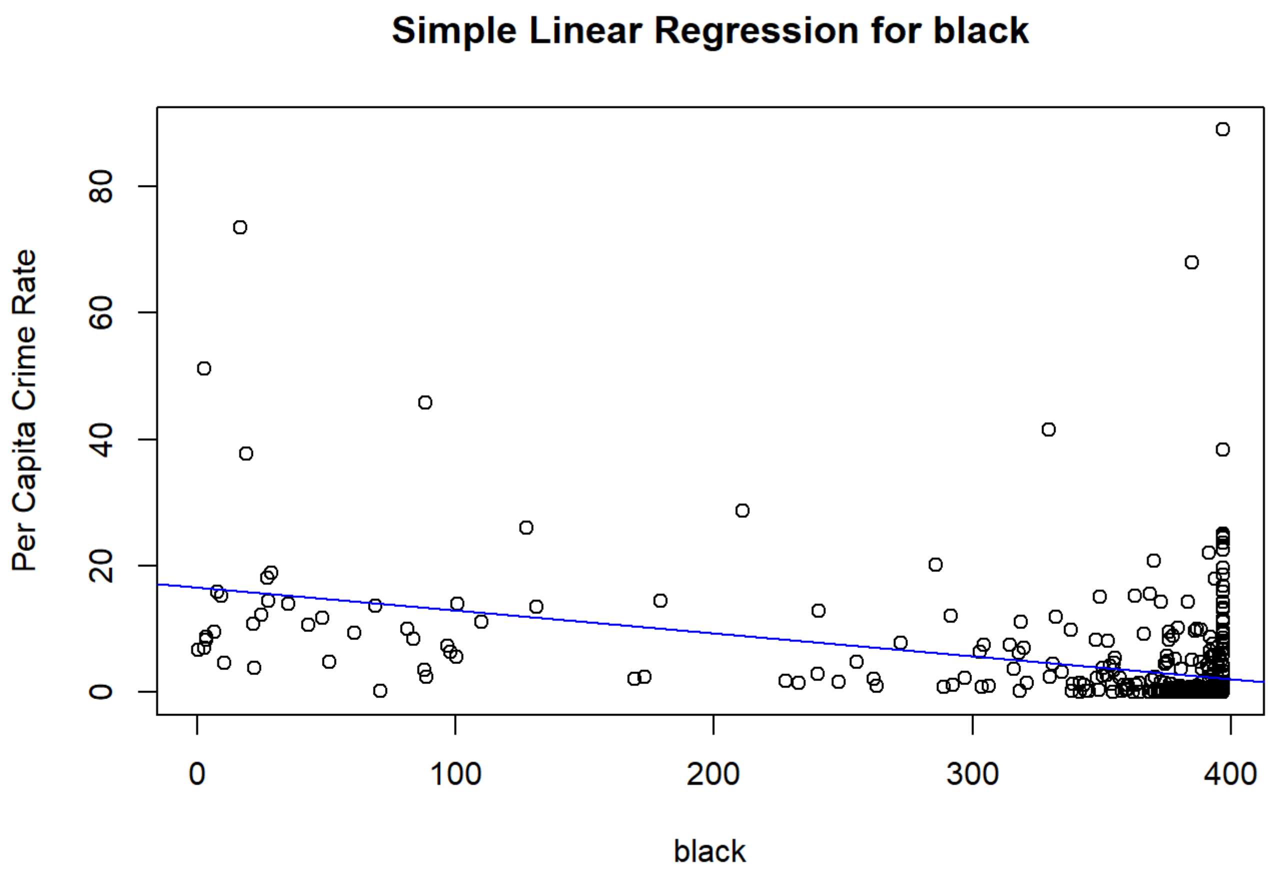



```
summary(lm.fit10)
```

```
##
## 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
```

According to the summary the p-value is low and regression coefficient is 1.1520. Means there is a positive relationship between ptratio and crim. Clearly, we can see that in the above plot.

```
lm.fit11 <- lm(crim ~ black, data = Boston)

plot(Boston$black , Boston$crim, xlab = "black", ylab = "Per Capita Crime Rate", main = "Simp
le Linear Regression for black")
abline(lm.fit11, col = "blue")
```

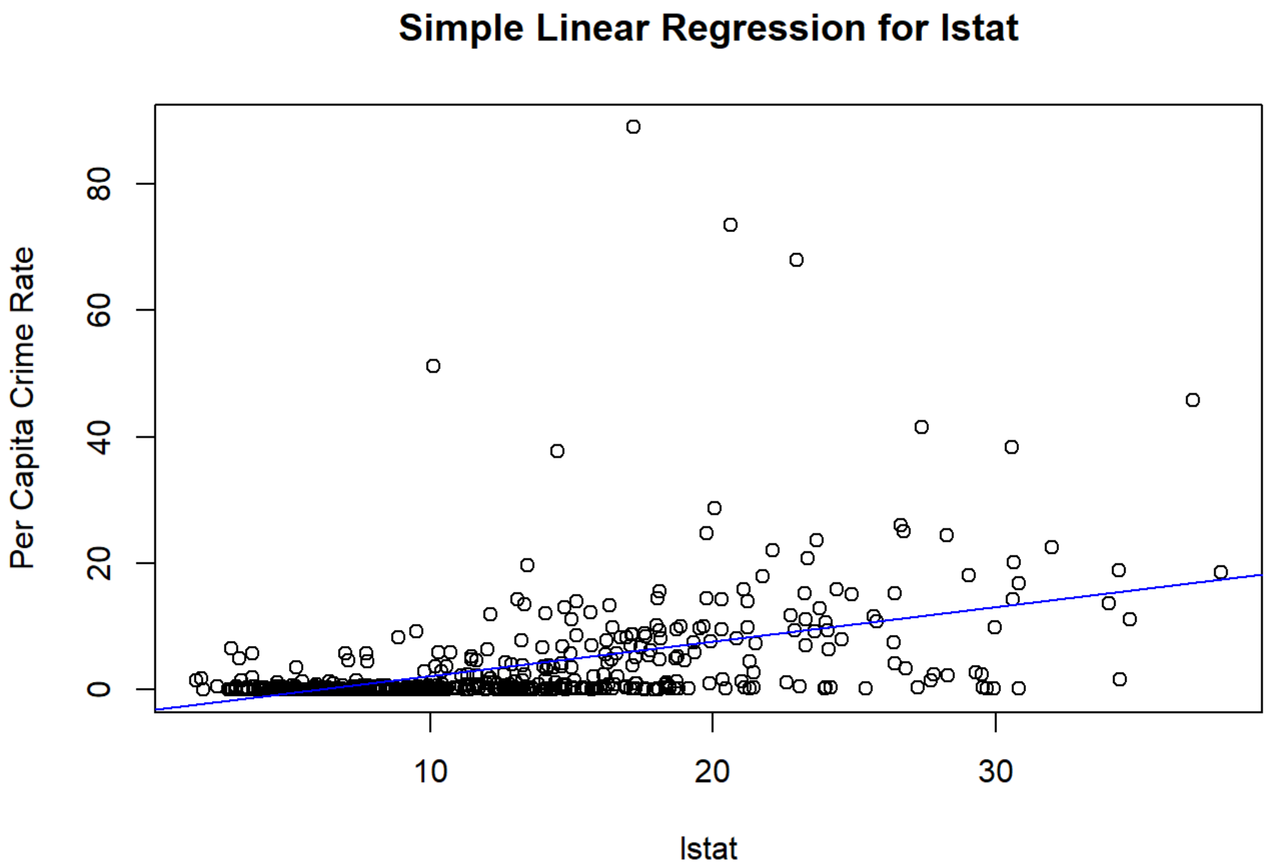



```
summary(lm.fit11)
```

```
##
## 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
```

According to the summary the p-value is low and regression coefficient is -0.036280. Means there is a negative relationship between black and crim. Clearly, we can see that in the above plot.

```
lm.fit12 <- lm(crim ~ lstat, data = Boston)

plot(Boston$lstat , Boston$crim, xlab = "lstat", ylab = "Per Capita Crime Rate", main = "Simp
le Linear Regression for lstat")
abline(lm.fit12, col = "blue")
```

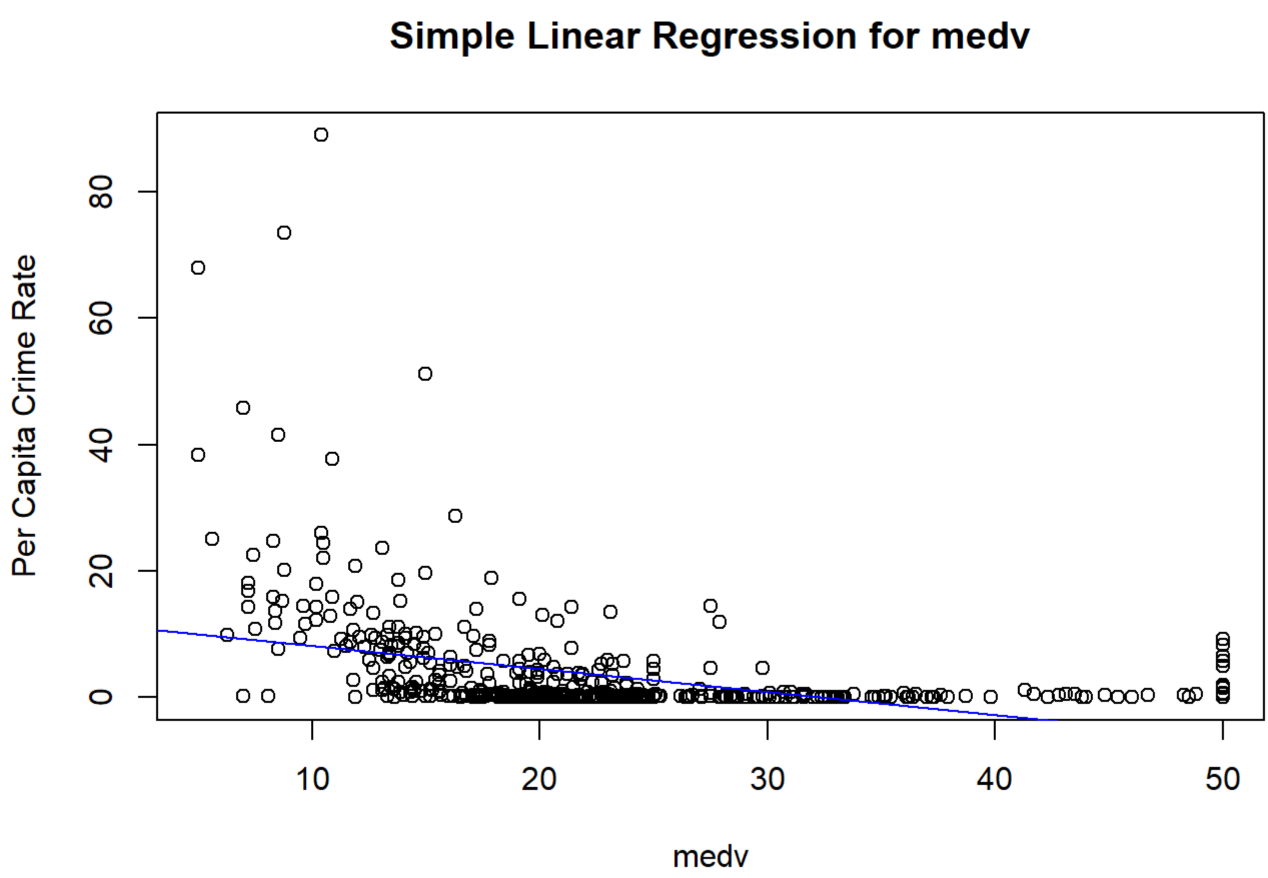



```
summary(lm.fit12)
```

```
##
## 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
```

According to the summary the p-value is low and regression coefficient is 0.54880. Means there is a positive relationship between lstat and crim. Clearly, we can see that in the above plot.

```
lm.fit13 <- lm(crim ~ medv, data = Boston)

plot(Boston$medv , Boston$crim, xlab = "medv", ylab = "Per Capita Crime Rate", main = "Simple
Linear Regression for medv")
abline(lm.fit13, col = "blue")
```



```
summary(lm.fit13)
```

```
##
## 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
```

According to the summary the p-value is low and regression coefficient is -0.36316. Means there is a negative relationship between medv and crim. Clearly, we can see that in the above plot.

b) Fit a multiple regression model to predict the response using all of the predictors. Describe your results. For which predictors can we reject the null hypothesis H0 : βj = 0?

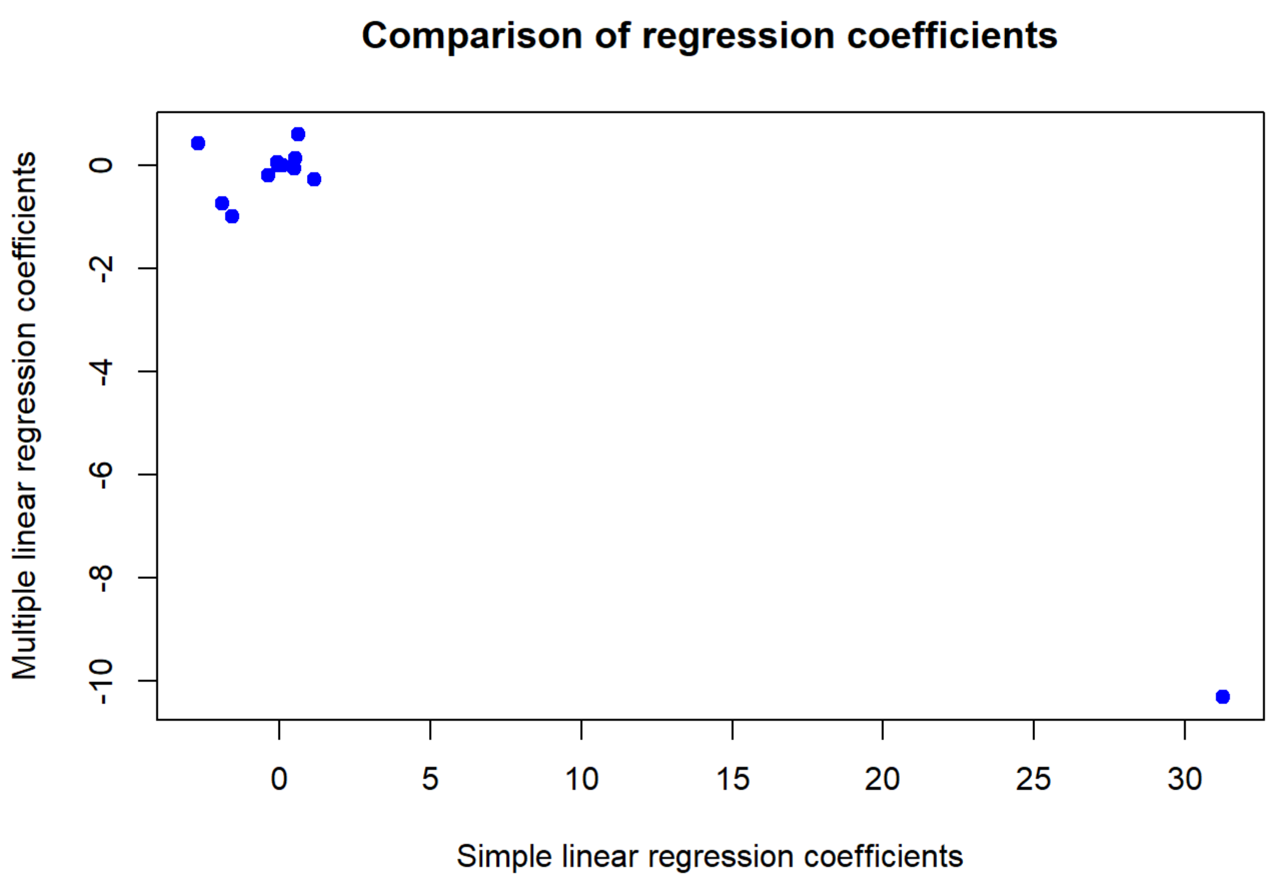
```
lm.mlfit <- lm(crim ~., data = Boston)
summary(lm.mlfit)
```

```
##
## 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
```

```
#### Here according to the p-values(Low) only few variables are significant they are "med
v","rad","dis", "black" and "zn". So, we can reject the null hypothesis for these variables.

#### Regression coefficient for zn is 0.0457100. Means there is a positive relationship betwe
en zn and crim.

#### Regression coefficient for dis is -1.0122467. Means there is a negative relationship bet
ween dis and crim.

#### Regression coefficient for rad is 0.6124653. Means there is a positive relationship betw
een rad and crim.

#### Regression coefficient for black is -0.007538. Means there is a negative relationship be
tween black and crim.

#### Regression coefficient for medv is -0.2200564. Means there is a negative relationship be
tween medv and crim.
```

c) How do your results from (a) compare to your results from (b)? Create a plot displaying the univariate regression coefficients from (a) on the x-axis, and the multiple regression coefficients from (b) on the y-axis. That is, each predictor is displayed as a single point in the plot. Its coefficient in a simple linear regression model is shown on the x-axis, and its coefficient estimate in the multiple linear regression model is shown on the y-axis.

```
### Coefficients of linear regression for each predictor.

linear_coefficients =
    c(coefficients(lm.fit1)[2],
      coefficients(lm.fit2)[2],
      coefficients(lm.fit3)[2],
      coefficients(lm.fit4)[2],
      coefficients(lm.fit5)[2],
      coefficients(lm.fit6)[2],
      coefficients(lm.fit7)[2],
      coefficients(lm.fit8)[2],
      coefficients(lm.fit9)[2],
      coefficients(lm.fit10)[2],
      coefficients(lm.fit11)[2],
      coefficients(lm.fit12)[2],
      coefficients(lm.fit13)[2])


### Coefficients of multiple regression. Excluding intercept.

multi_coefficients <- coefficients(lm.mlfit)[2:14]


plot(linear_coefficients , multi_coefficients, xlab = "Simple linear regression coefficient
s", ylab = "Multiple linear regression coefficients", main = "Comparison of regression coeffi
cients", col = 'blue', pch = 19)
```



```
### Regression coefficient for 'nox' in simple linear regression is 31.24853120 and in multip
le linear regression is -9.9575865471
```

d) Is there evidence of non-linear association between any of the predictors and the response? To answer this question, for each predictor X, fit a model of the form Y = β0 + β1X + β2(X^2) + β3(X^3) + ϵ.

```
lm.cubfit1 <- lm(crim ~ zn + I(zn^2) + I(zn^3), data = Boston)
summary(lm.cubfit1)
```

```
##
## Call:
## lm(formula = crim ~ zn + I(zn^2) + I(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)  4.846e+00  4.330e-01  11.192  < 2e-16 ***
## zn          -3.322e-01  1.098e-01  -3.025  0.00261 **
## I(zn^2)      6.483e-03  3.861e-03   1.679  0.09375 .
## I(zn^3)     -3.776e-05  3.139e-05  -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
```

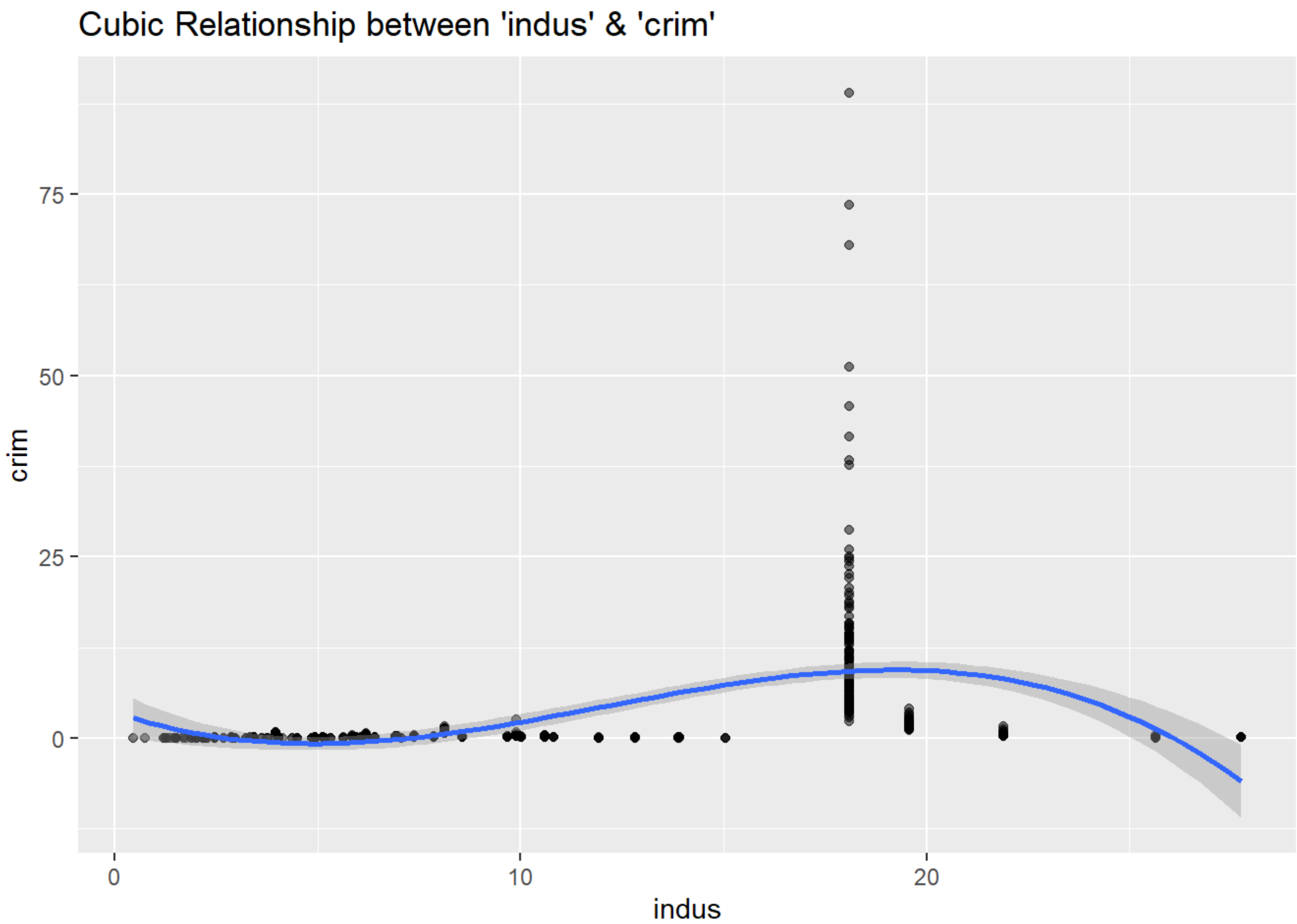
As per the p-values of t statistic the relationship between zn and crim is linear.

```
lm.cubfit2 <- lm(crim ~ indus + I(indus^2) + I(indus^3), data = Boston)
summary(lm.cubfit2)
```

```
##
## Call:
## lm(formula = crim ~ indus + I(indus^2) + I(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)  3.6625683  1.5739833   2.327   0.0204 *
## indus       -1.9652129  0.4819901  -4.077 5.30e-05 ***
## I(indus^2)   0.2519373  0.0393221   6.407 3.42e-10 ***
## I(indus^3)  -0.0069760  0.0009567  -7.292 1.20e-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
```

As per the p-values of t statistic the relationship between indus and crim is cubic.

```
library(ggplot2)
ggplot(Boston, aes(x = indus, y = crim)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", formula = "y ~ x + I(x^2) + I(x^3)") +
  labs(title = "Cubic Relationship between 'indus' & 'crim'")
```

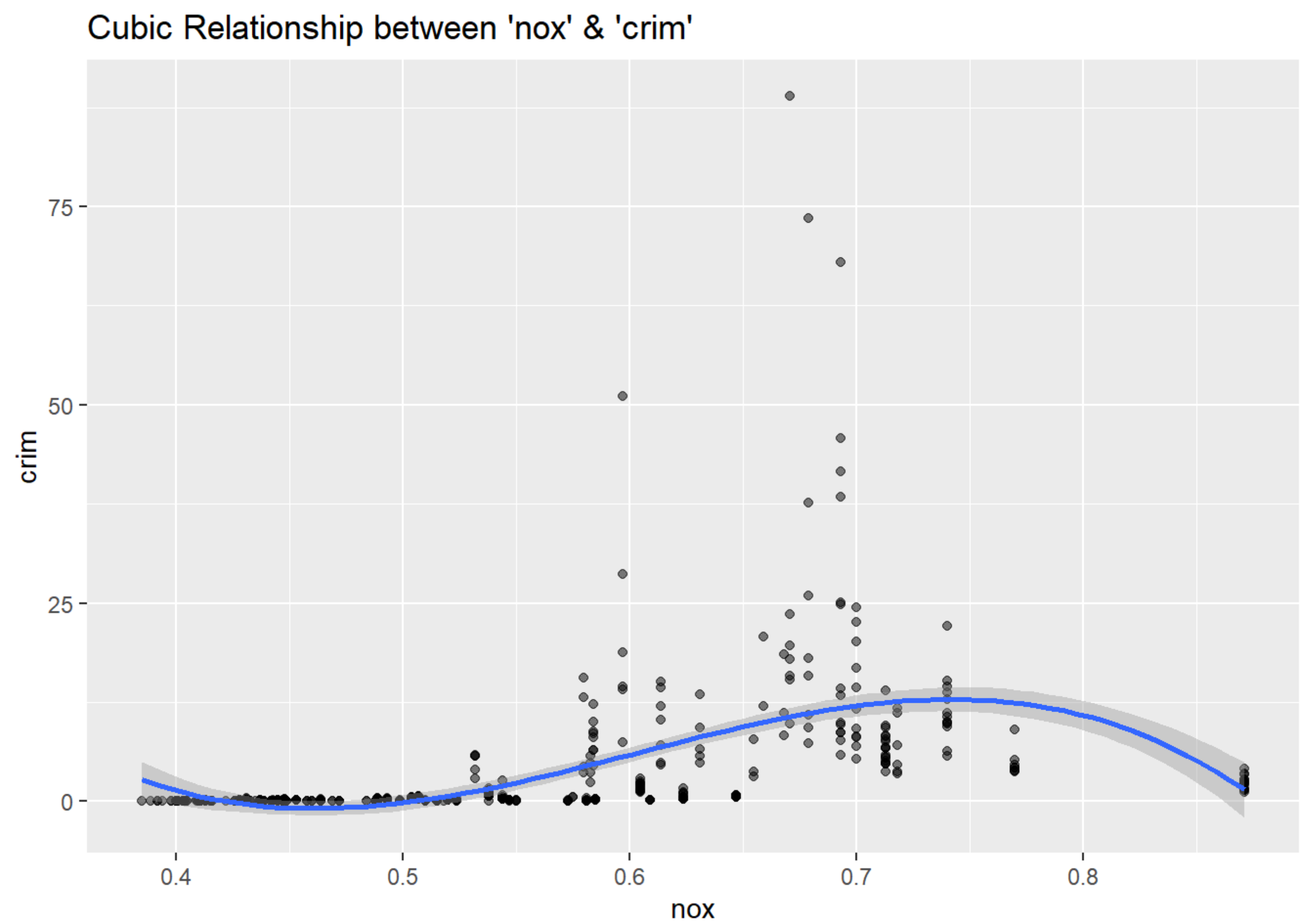


```
lm.cubfit3 <- lm(crim ~ nox + I(nox^2) + I(nox^3), data = Boston)
summary(lm.cubfit3)
```

```
##
## Call:
## lm(formula = crim ~ nox + I(nox^2) + I(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)   233.09      33.64   6.928 1.31e-11 ***
## nox         -1279.37     170.40  -7.508 2.76e-13 ***
## I(nox^2)     2248.54     279.90   8.033 6.81e-15 ***
## I(nox^3)    -1245.70     149.28  -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
```

As per the p-values of t statistic the relationship between nox and crim is cubic.

```
ggplot(Boston, aes(x = nox, y = crim)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", formula = "y ~ x + I(x^2) + I(x^3)") +
  labs(title = "Cubic Relationship between 'nox' & 'crim'")
```




```
lm.cubfit4 <- lm(crim ~ rm + I(rm^2) + I(rm^3), data = Boston)
summary(lm.cubfit4)
```

```
##
## Call:
## lm(formula = crim ~ rm + I(rm^2) + I(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) 112.6246    64.5172   1.746   0.0815 .
## rm          -39.1501    31.3115  -1.250   0.2118
## I(rm^2)       4.5509     5.0099   0.908   0.3641
## I(rm^3)      -0.1745     0.2637  -0.662   0.5086
## ---
## 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
```

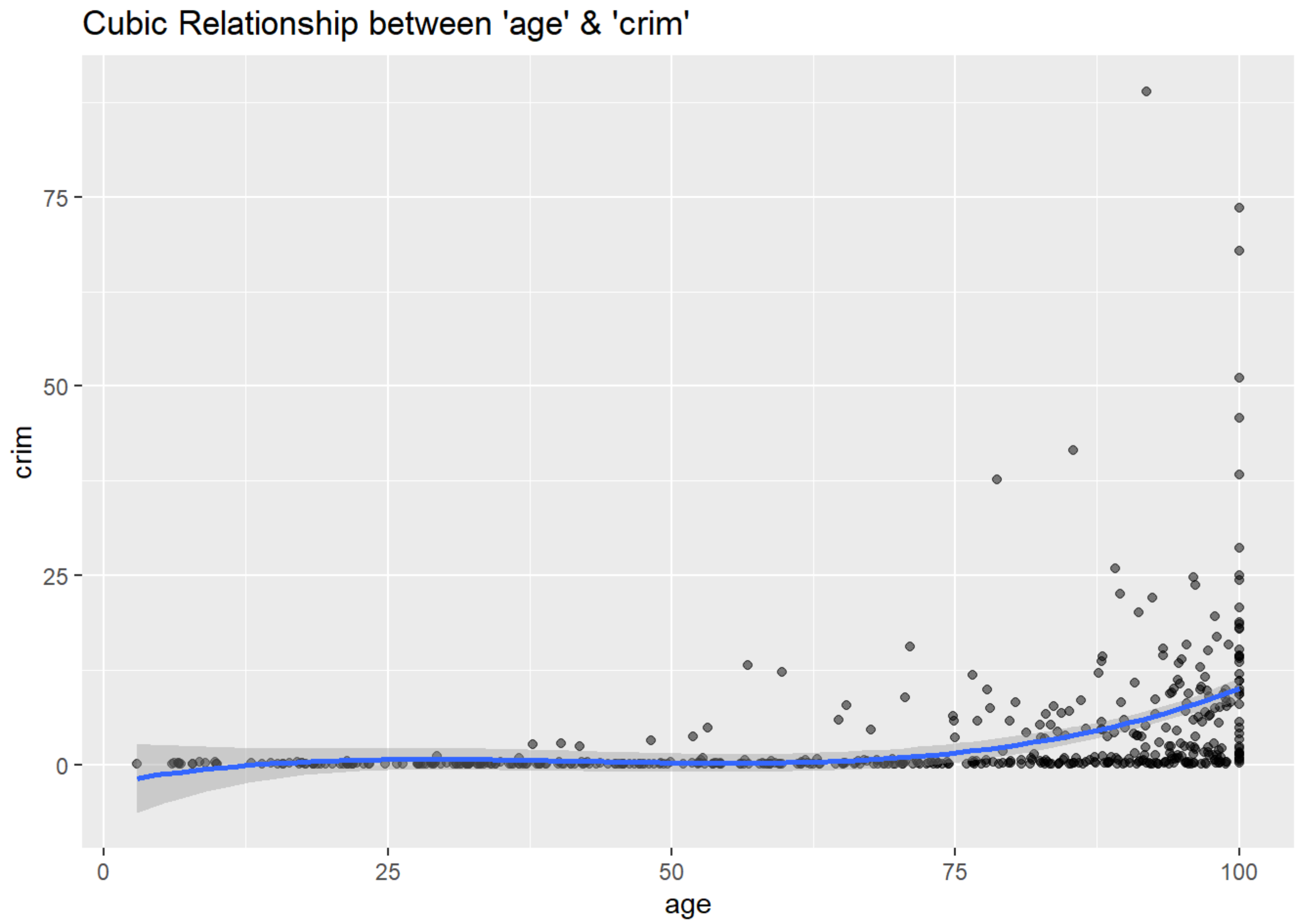
As per the p-values of t statistic there is no relationship between rm and crim.

```r
lm.cubfit5 <- lm(crim ~ age + I(age^2) + I(age^3), data = Boston)
summary(lm.cubfit5)
```

```
##
## Call:
## lm(formula = crim ~ age + I(age^2) + I(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) -2.549e+00  2.769e+00  -0.920  0.35780
## age          2.737e-01  1.864e-01   1.468  0.14266
## I(age^2)    -7.230e-03  3.637e-03  -1.988  0.04738 *
## I(age^3)     5.745e-05  2.109e-05   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
```

As per the p-values of t statistic the relationship between age and crim is cubic.

```r
ggplot(Boston, aes(x = age, y = crim)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", formula = "y ~ x + I(x^2) + I(x^3)") +
  labs(title = "Cubic Relationship between 'age' & 'crim'")
```

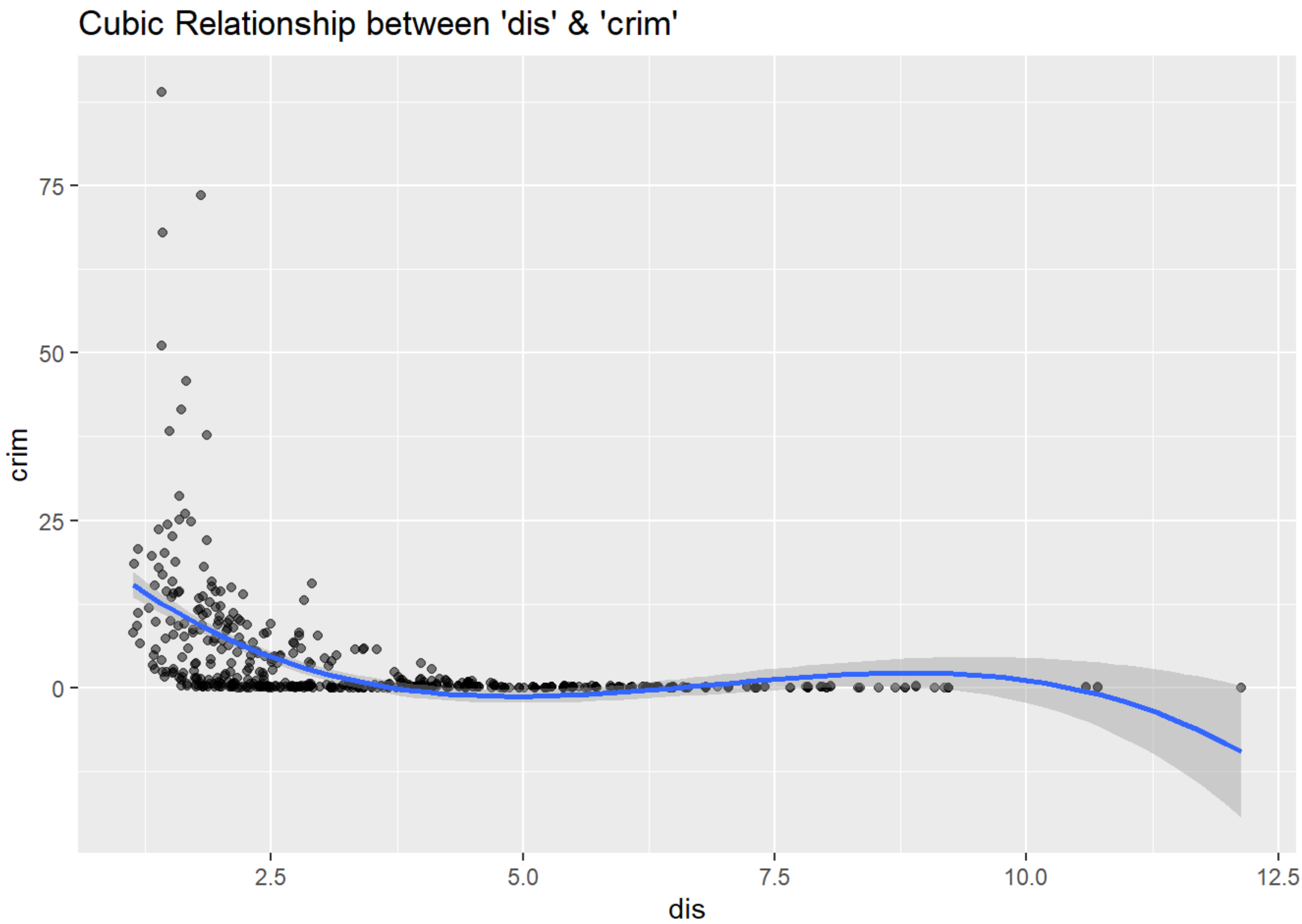



```
lm.cubfit6 <- lm(crim ~ dis + I(dis^2) + I(dis^3), data = Boston)
summary(lm.cubfit6)
```

```
##
## Call:
## lm(formula = crim ~ dis + I(dis^2) + I(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)  30.0476     2.4459  12.285  < 2e-16 ***
## dis         -15.5543     1.7360  -8.960  < 2e-16 ***
## I(dis^2)      2.4521     0.3464   7.078 4.94e-12 ***
## I(dis^3)     -0.1186     0.0204  -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
```

As per the p-values of t statistic the relationship between dis and crim is cubic.

```
ggplot(Boston, aes(x = dis, y = crim)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", formula = "y ~ x + I(x^2) + I(x^3)") +
  labs(title = "Cubic Relationship between 'dis' & 'crim'")
```




```
lm.cubfit7 <- lm(crim ~ rad + I(rad^2) + I(rad^3), data = Boston)
summary(lm.cubfit7)
```

```
##
## Call:
## lm(formula = crim ~ rad + I(rad^2) + I(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) -0.605545   2.050108  -0.295    0.768
## rad          0.512736   1.043597   0.491    0.623
## I(rad^2)    -0.075177   0.148543  -0.506    0.613
## I(rad^3)     0.003209   0.004564   0.703    0.482
##
## 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
```

As per the p-values of t statistic there is no relationship between rad and crim.

```
lm.cubfit8 <- lm(crim ~ tax + I(tax^2) + I(tax^3), data = Boston)
summary(lm.cubfit8)
```

```
##
## Call:
## lm(formula = crim ~ tax + I(tax^2) + I(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)  1.918e+01  1.180e+01   1.626    0.105
## tax         -1.533e-01  9.568e-02  -1.602    0.110
## I(tax^2)     3.608e-04  2.425e-04   1.488    0.137
## I(tax^3)    -2.204e-07  1.889e-07  -1.167    0.244
##
## 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
```

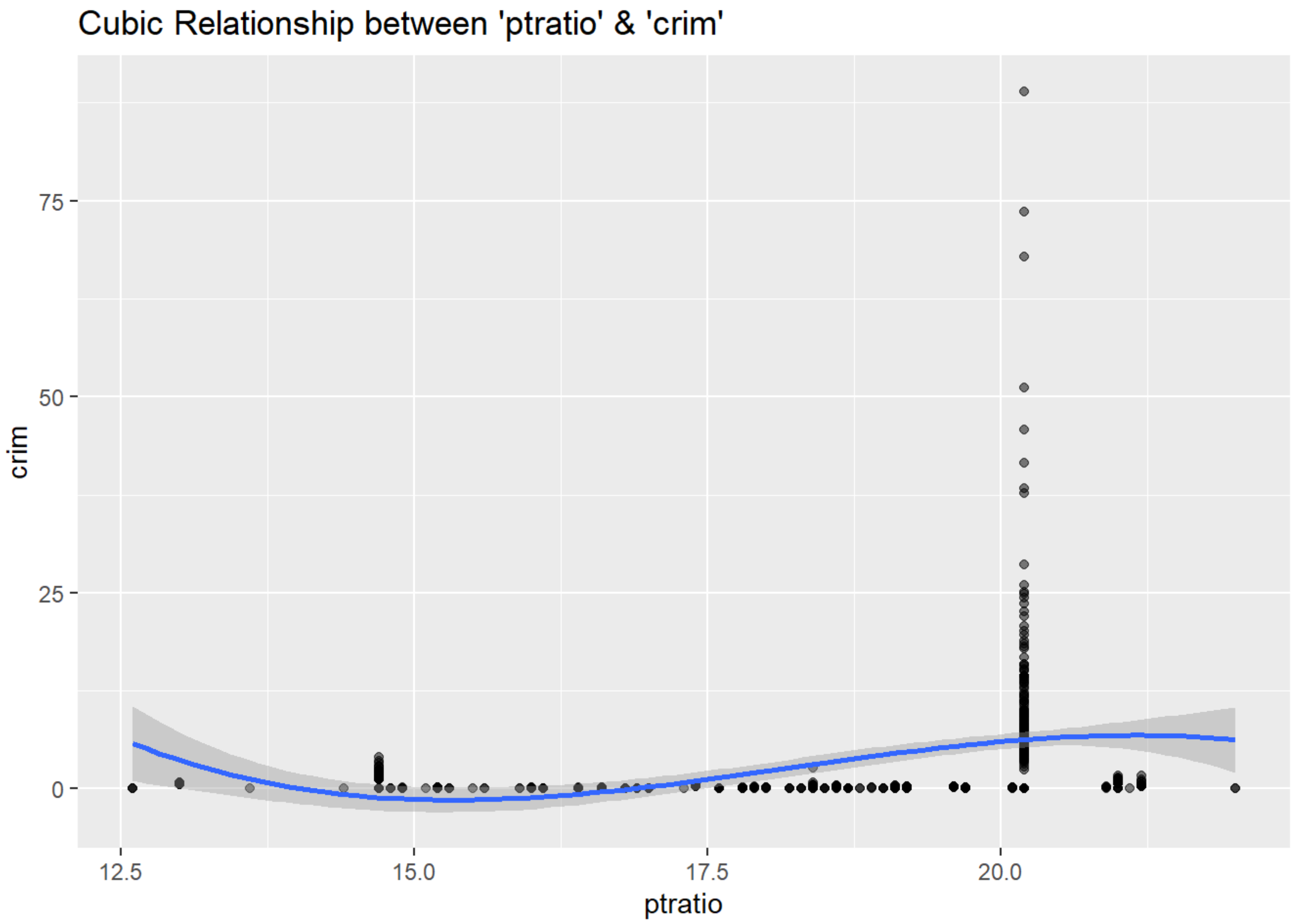
As per the p-values of t statistic there is no relationship between tax and crim.

```
lm.cubfit9 <- lm(crim ~ ptratio + I(ptratio^2) + I(ptratio^3), data = Boston)
summary(lm.cubfit9)
```

```
##
## Call:
## lm(formula = crim ~ ptratio + I(ptratio^2) + I(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)  477.18405  156.79498   3.043  0.00246 **
## ptratio      -82.36054   27.64394  -2.979  0.00303 **
## I(ptratio^2)   4.63535    1.60832   2.882  0.00412 **
## I(ptratio^3)  -0.08476    0.03090  -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
```

As per the p-values of t statistic the relationship between ptratio and crim is cubic.

```
ggplot(Boston, aes(x = ptratio, y = crim)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", formula = "y ~ x + I(x^2) + I(x^3)") +
  labs(title = "Cubic Relationship between 'ptratio' & 'crim'")
```




```
lm.cubfit10 <- lm(crim ~ black + I(black^2) + I(black^3), data = Boston)
summary(lm.cubfit10)
```

```
##
## Call:
## lm(formula = crim ~ black + I(black^2) + I(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)  1.826e+01  2.305e+00   7.924  1.5e-14 ***
## black       -8.356e-02  5.633e-02  -1.483    0.139
## I(black^2)   2.137e-04  2.984e-04   0.716    0.474
## I(black^3)  -2.652e-07  4.364e-07  -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
```

As per the p-values of t statistic there is no relationship between black and crim.

```
lm.cubfit11 <- lm(crim ~ lstat + I(lstat^2) + I(lstat^3), data = Boston)
summary(lm.cubfit11)
```

```
##
## Call:
## lm(formula = crim ~ lstat + I(lstat^2) + I(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)  1.2009656  2.0286452   0.592   0.5541
## lstat       -0.4490656  0.4648911  -0.966   0.3345
## I(lstat^2)   0.0557794  0.0301156   1.852   0.0646 .
## I(lstat^3)  -0.0008574  0.0005652  -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
```

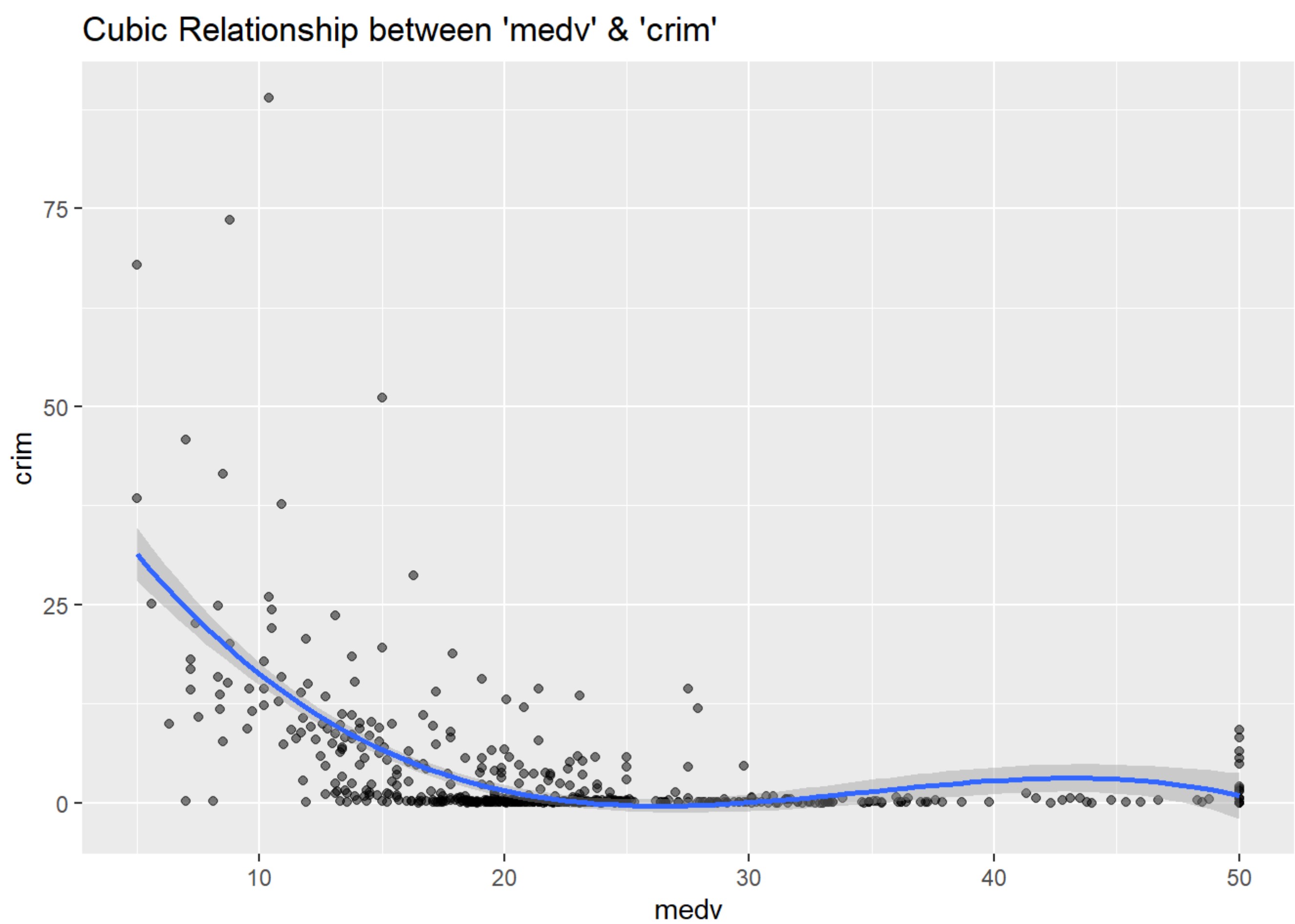
As per the p-values of t statistic there is no relationship between lstat and crim.

```
lm.cubfit12 <- lm(crim ~ medv + I(medv^2) + I(medv^3), data = Boston)
summary(lm.cubfit12)
```

```
##
## Call:
## lm(formula = crim ~ medv + I(medv^2) + I(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) 53.1655381  3.3563105  15.840  < 2e-16 ***
## medv        -5.0948305  0.4338321 -11.744  < 2e-16 ***
## I(medv^2)    0.1554965  0.0171904   9.046  < 2e-16 ***
## I(medv^3)   -0.0014901  0.0002038  -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
```

As per the p-values of t statistic the relationship between medv and crim is cubic.

```
ggplot(Boston, aes(x = medv, y = crim)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", formula = "y ~ x + I(x^2) + I(x^3)") +
  labs(title = "Cubic Relationship between 'medv' & 'crim'")
```



Question 4) Consider the real-estate data provided in the regression computational lab. Apply best, forward, and backwards subset selection to the real estate data. Compare the performance of the methods, and the variables that were selected in the "optimal models" using test/training, BIC and Cp.

```
setwd("D:/Buffalo/files")

dats <- read.delim("Real estate.csv", sep = ",", header = TRUE)
head(dats)
```

```
##   No X1.transaction.date X2.house.age X3.distance.to.the.nearest.MRT.station
## 1  1            2012.917         32.0                               84.87882
## 2  2            2012.917         19.5                              306.59470
## 3  3            2013.583         13.3                              561.98450
## 4  4            2013.500         13.3                              561.98450
## 5  5            2012.833          5.0                              390.56840
## 6  6            2012.667          7.1                             2175.03000
##   X4.number.of.convenience.stores X5.latitude X6.longitude
## 1                              10    24.98298     121.5402
## 2                               9    24.98034     121.5395
## 3                               5    24.98746     121.5439
## 4                               5    24.98746     121.5439
## 5                               5    24.97937     121.5425
## 6                               3    24.96305     121.5125
##   Y.house.price.of.unit.area
## 1                       37.9
## 2                       42.2
## 3                       47.3
## 4                       54.8
## 5                       43.1
## 6                       32.1
```

```
dim(dats)
```

```
## [1] 414   8
```

```
# eliminating the first column (index)

dats <- dats[,-1]

dim(dats) # 414 x 7
```

```
## [1] 414   7
```

```
# redefining the variable names

colnames(dats) <- c("Trans.date", "House.Age", "Dist.2.Transp", "No.stores", "Lat", "Long",
"PriceUnit")
head(dats)
```

```
##   Trans.date House.Age Dist.2.Transp No.stores      Lat     Long PriceUnit
## 1   2012.917      32.0      84.87882        10 24.98298 121.5402      37.9
## 2   2012.917      19.5     306.59470         9 24.98034 121.5395      42.2
## 3   2013.583      13.3     561.98450         5 24.98746 121.5439      47.3
## 4   2013.500      13.3     561.98450         5 24.98746 121.5439      54.8
## 5   2012.833       5.0     390.56840         5 24.97937 121.5425      43.1
## 6   2012.667       7.1    2175.03000         3 24.96305 121.5125      32.1
```

```
apply(dats, 2, 'class')  ##To check class of variables.
```

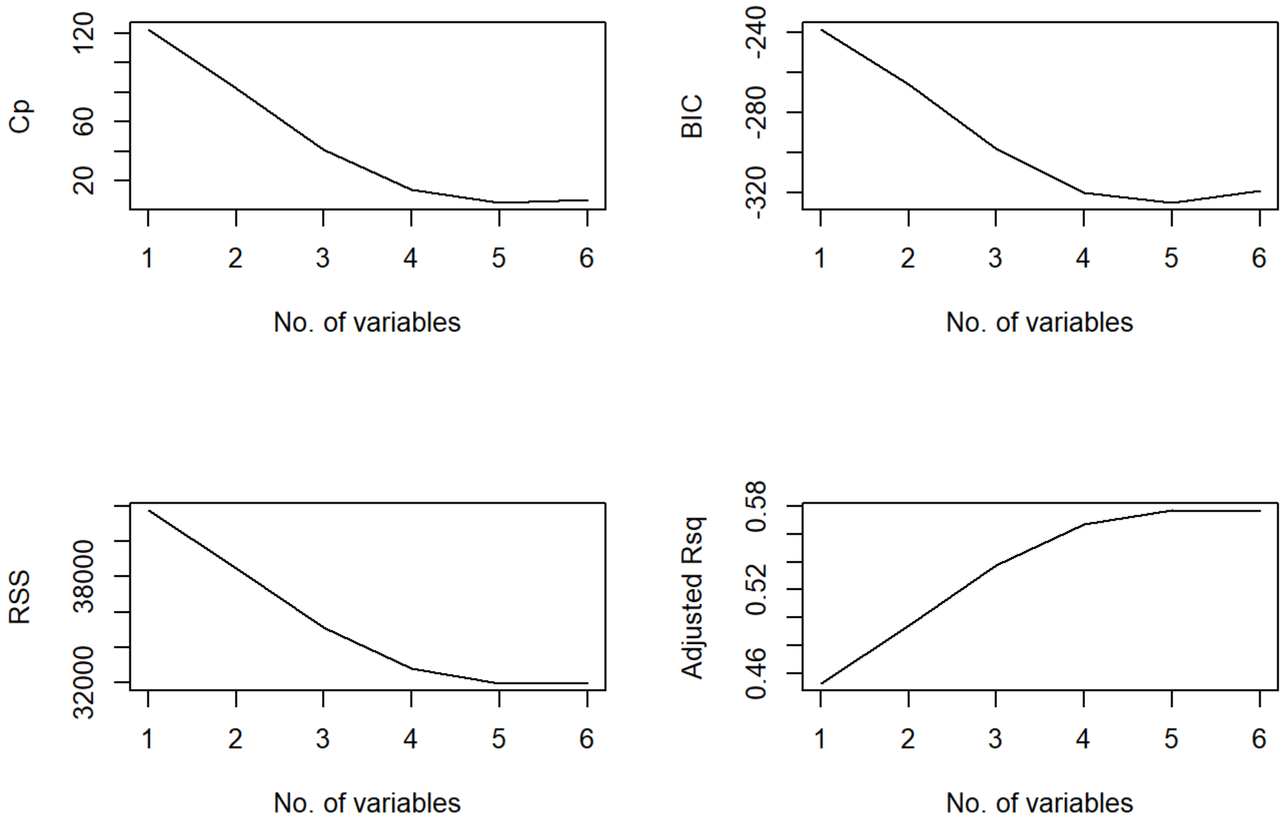
```
##    Trans.date     House.Age Dist.2.Transp     No.stores           Lat
##     "numeric"     "numeric"     "numeric"     "numeric"     "numeric"
##          Long     PriceUnit
##     "numeric"     "numeric"
```

Best subset selection:

```
# Performing best subset selection on the data

library(leaps)

regfit.full <- regsubsets(PriceUnit~., data = dats, nbest = 1, nvmax = 6, method = "exhaustiv
e")
my_sum <- summary(regfit.full)
names(my_sum)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

```
# plot model selection measures
par(mfrow = c(2,2))
plot(my_sum$cp, xlab = "No. of variables", ylab = "Cp", type = "l")
plot(my_sum$bic, xlab = "No. of variables", ylab = "BIC", type = "l")
plot(my_sum$rss, xlab = "No. of variables", ylab = "RSS", type = "l")
plot(my_sum$adjr2, xlab = "No. of variables", ylab = "Adjusted Rsq", type = "l")
```




```
# identify the optimal models using model selection measures for best subset selection
which.min(my_sum$cp) #5 var
```

```
## [1] 5
```

```
which.min(my_sum$bic) #5 var
```

```
## [1] 5
```

```
which.min(my_sum$rss) #6 var
```

```
## [1] 6
```

```
which.max(my_sum$adjr2) #5 var
```

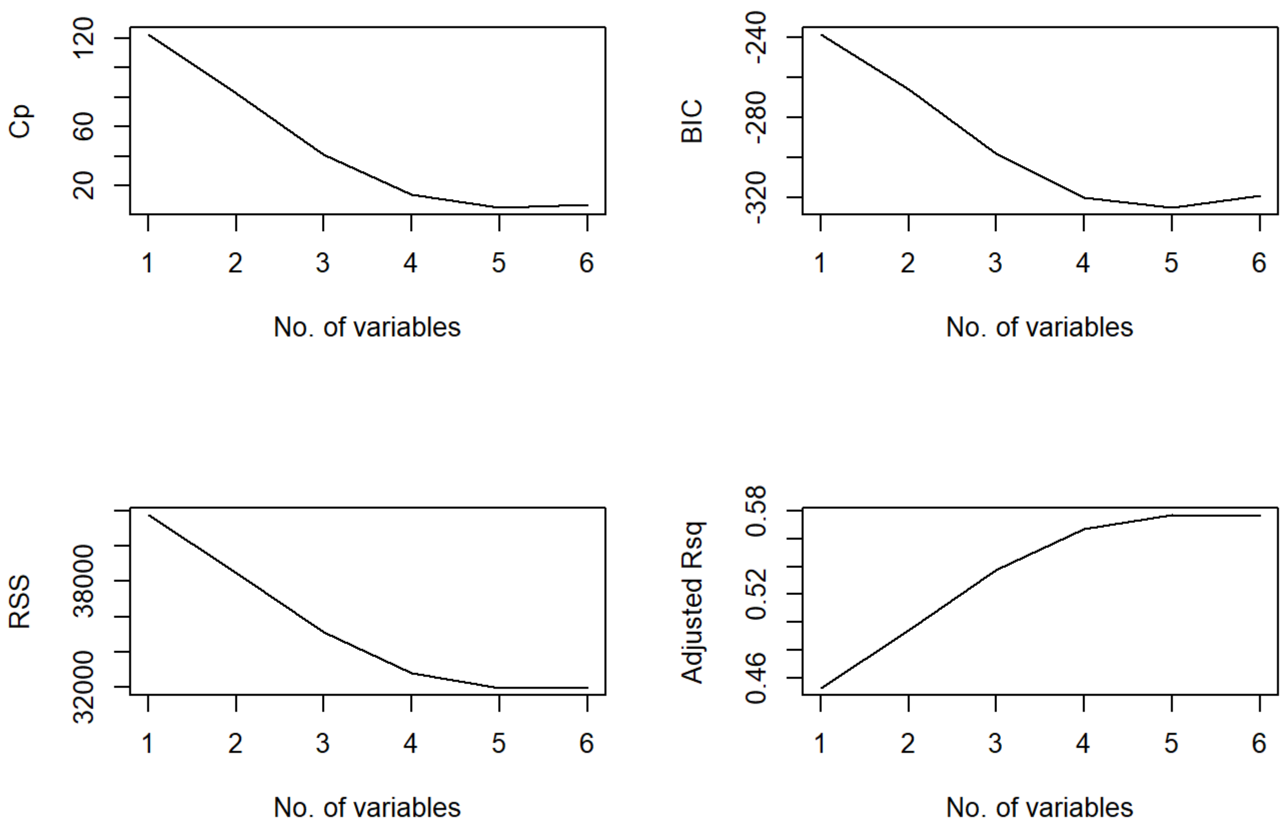
```
## [1] 5
```

Forward subset selection:

```
# Performing forward subset selection on the data

regfit.fwd <- regsubsets(PriceUnit~., data = dats, nbest = 1, nvmax = 6, method = "forward")

my_sum_fwd <- summary(regfit.fwd)
names(my_sum_fwd)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

```
# plot model selection measures
par(mfrow = c(2,2))
plot(my_sum_fwd$cp, xlab = "No. of variables", ylab = "Cp", type = "l")
plot(my_sum_fwd$bic, xlab = "No. of variables", ylab = "BIC", type = "l")
plot(my_sum_fwd$rss, xlab = "No. of variables", ylab = "RSS", type = "l")
plot(my_sum_fwd$adjr2, xlab = "No. of variables", ylab = "Adjusted Rsq", type = "l")
```




```
# identify the optimal models using model selection measures for forward subset selection

which.min(my_sum_fwd$cp) #5 var
```

```
## [1] 5
```

```
which.min(my_sum_fwd$bic) #5 var
```

```
## [1] 5
```

```
which.min(my_sum_fwd$rss) #6 var
```

```
## [1] 6
```

```
which.max(my_sum_fwd$adjr2) #5 var
```

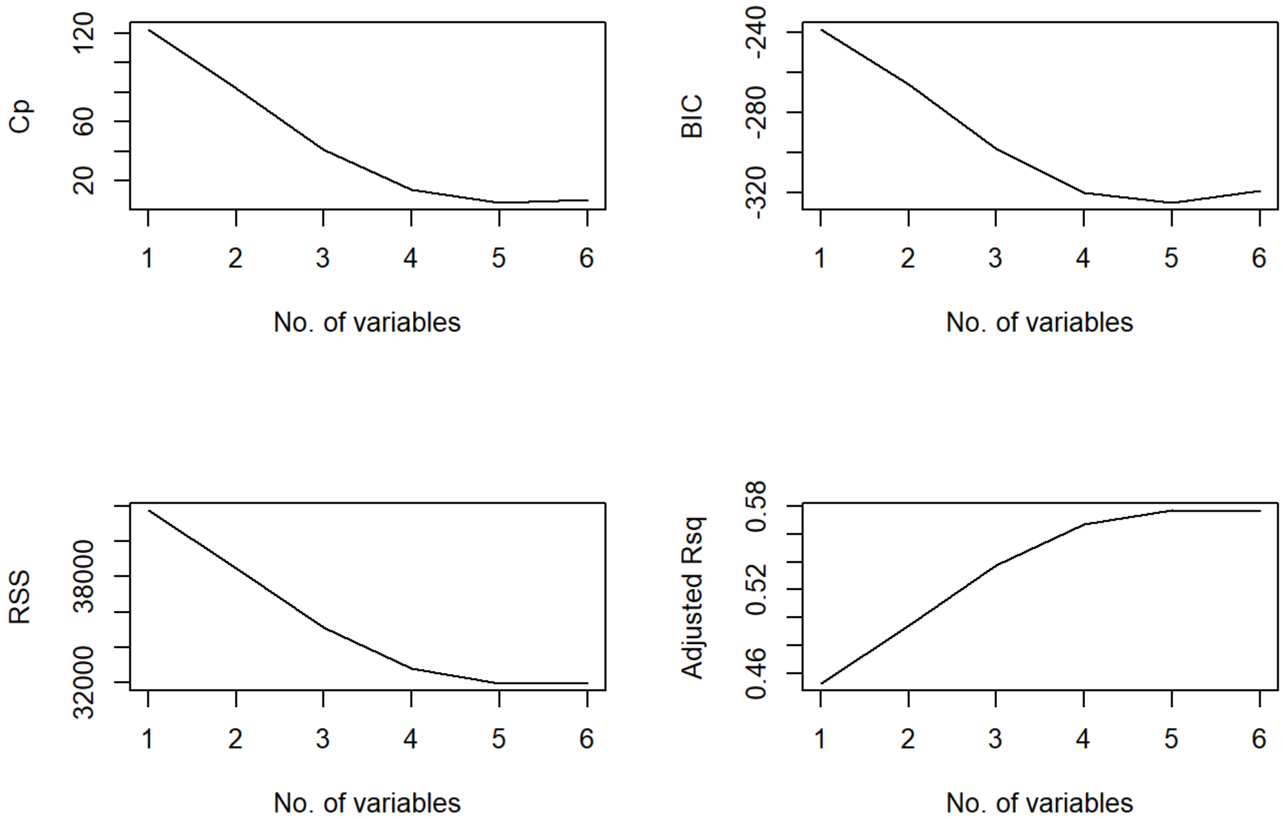
```
## [1] 5
```

Backward subset selection:

```
# Performing backward subset selection on the data

regfit.bwd <- regsubsets(PriceUnit~., data = dats, nbest = 1, nvmax = 6, method = "backward")

my_sum_bwd <- summary(regfit.bwd)

names(my_sum_bwd)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

```
# plot model selection measures
par(mfrow = c(2,2))
plot(my_sum_bwd$cp, xlab = "No. of variables", ylab = "Cp", type = "l")
plot(my_sum_bwd$bic, xlab = "No. of variables", ylab = "BIC", type = "l")
plot(my_sum_bwd$rss, xlab = "No. of variables", ylab = "RSS", type = "l")
plot(my_sum_bwd$adjr2, xlab = "No. of variables", ylab = "Adjusted Rsq", type = "l")
```




```
# identify the optimal models using model selection measures for backward subset selection

which.min(my_sum_bwd$cp) #5 var
```

```
## [1] 5
```

```
which.min(my_sum_bwd$bic) #5 var
```

```
## [1] 5
```

```
which.min(my_sum_bwd$rss) #6 var
```

```
## [1] 6
```

```
which.max(my_sum_bwd$adjr2) #5 var
```

```
## [1] 5
```

```
# examine the best "p" variables models

my_sum$outmat
```

```
##          Trans.date House.Age Dist.2.Transp No.stores Lat Long
## 1  ( 1 ) " "        " "       "*"           " "       " " " " 
## 2  ( 1 ) " "        " "       "*"           "*"       " " " " 
## 3  ( 1 ) " "        "*"       "*"           "*"       " " " " 
## 4  ( 1 ) " "        "*"       "*"           "*"       "*" " " 
## 5  ( 1 ) "*"        "*"       "*"           "*"       "*" " " 
## 6  ( 1 ) "*"        "*"       "*"           "*"       "*" "*"
```

```
my_sum_fwd$outmat
```

```
##          Trans.date House.Age Dist.2.Transp No.stores Lat Long
## 1  ( 1 ) " "        " "       "*"           " "       " " " " 
## 2  ( 1 ) " "        " "       "*"           "*"       " " " " 
## 3  ( 1 ) " "        "*"       "*"           "*"       " " " " 
## 4  ( 1 ) " "        "*"       "*"           "*"       "*" " " 
## 5  ( 1 ) "*"        "*"       "*"           "*"       "*" " " 
## 6  ( 1 ) "*"        "*"       "*"           "*"       "*" "*"
```

```
my_sum_bwd$outmat
```

```
##          Trans.date House.Age Dist.2.Transp No.stores Lat Long
## 1  ( 1 ) " "        " "       "*"           " "       " " " " 
## 2  ( 1 ) " "        " "       "*"           "*"       " " " " 
## 3  ( 1 ) " "        "*"       "*"           "*"       " " " " 
## 4  ( 1 ) " "        "*"       "*"           "*"       "*" " " 
## 5  ( 1 ) "*"        "*"       "*"           "*"       "*" " " 
## 6  ( 1 ) "*"        "*"       "*"           "*"       "*" "*"
```

```
my_sum$outmat[3,]
```

```
##    Trans.date     House.Age Dist.2.Transp     No.stores           Lat 
##           " "           "*"           "*"           "*"           " " 
##          Long 
##           " "
```

```
my_sum_fwd$outmat[3,]
```

```
##    Trans.date     House.Age Dist.2.Transp     No.stores           Lat 
##           " "           "*"           "*"           "*"           " " 
##          Long 
##           " "
```

```
my_sum_bwd$outmat[3,]
```

```
##    Trans.date     House.Age Dist.2.Transp     No.stores           Lat 
##           " "           "*"           "*"           "*"           " " 
##          Long 
##           " "
```

```
coef(regfit.full, 3)
```

```
##   (Intercept)     House.Age Dist.2.Transp     No.stores
##   42.97728621   -0.25285583   -0.00537913    1.29744248
```

```
coef(regfit.fwd, 3)
```

```
##   (Intercept)     House.Age Dist.2.Transp     No.stores
##   42.97728621   -0.25285583   -0.00537913    1.29744248
```

```
coef(regfit.bwd, 3)
```

```
##   (Intercept)     House.Age Dist.2.Transp     No.stores
##   42.97728621   -0.25285583   -0.00537913    1.29744248
```

Minimum test error for Best subset selection using training and test data:

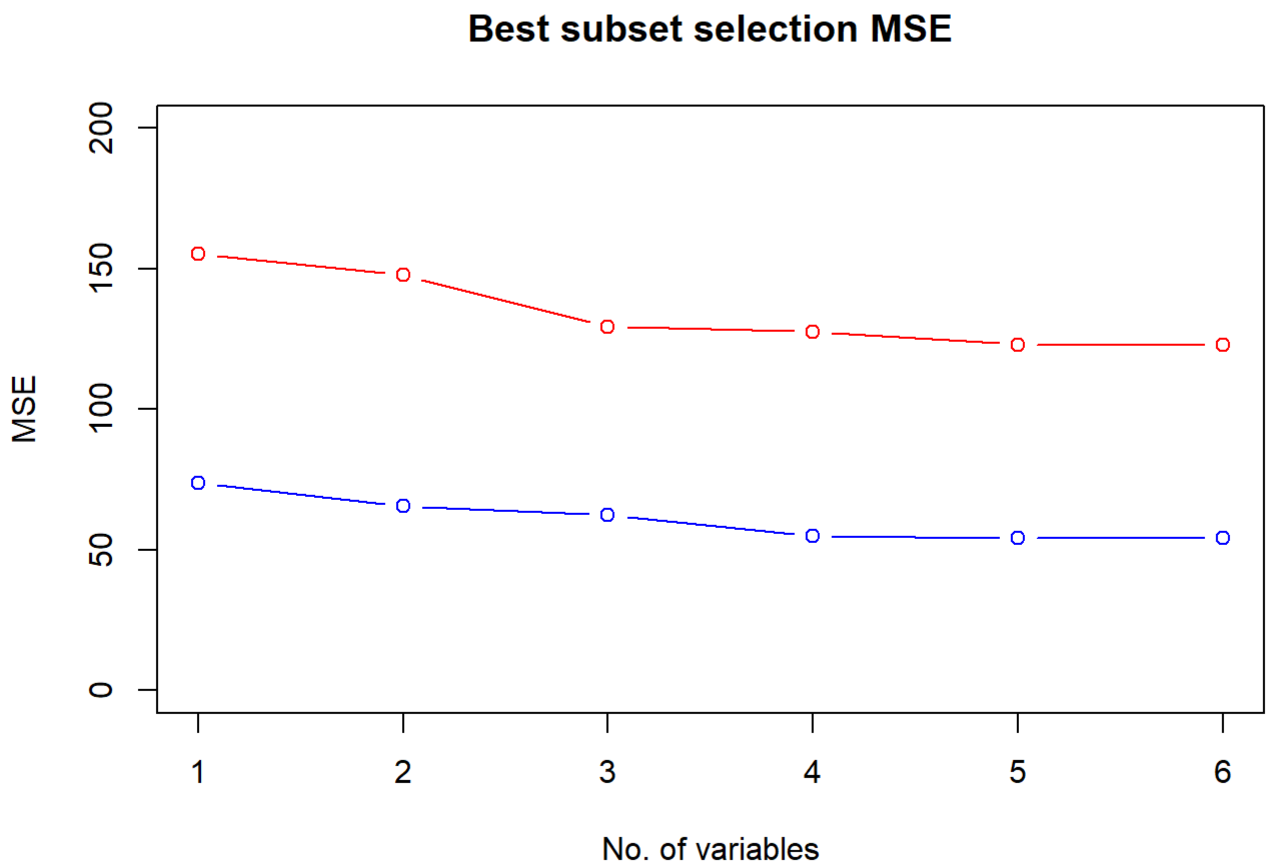
```
###  best subset selection:

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[,xvars]%*%coefi
}

# creating test and training data

set.seed(123)
train_indis <- sample(c(1:length(dats[,1])), size = 2/3*length(dats[,1]), replace = FALSE)

train = dats[train_indis, ]
dim(train)
```

```
## [1] 276   7
```

```
test = dats[-train_indis, ]
dim(test)
```

```
## [1] 138   7
```

```
y_true_train = train$PriceUnit
y_true_test = test$PriceUnit

# create objects to store error

train_err_store <- matrix(rep(NA, 6))
test_err_store <- matrix(rep(NA, 6))
regfit.full <- regsubsets(PriceUnit~., data = dats, nbest = 1, nvmax = 6, method = "exhaustiv
e") # perform subset selection

for (i in 1:6){
    # make the predictions
    y_hat_train = predict(regfit.full, newdata = train, id = i)
    y_hat_test = predict(regfit.full, newdata = test, id = i)

    # compare the prediction with the true
    train_err_store[i] = (1/length(y_true_train))*sum((y_true_train-y_hat_train)^2)
    test_err_store[i] = (1/length(y_true_test))*sum((y_true_test-y_hat_test)^2)

}

plot(train_err_store, col = "blue", type = "b", xlab = "No. of variables", ylab = "MSE", ylim
= c(0,200), main="Best subset selection MSE")
lines(test_err_store, col = "red", type = "b")
```

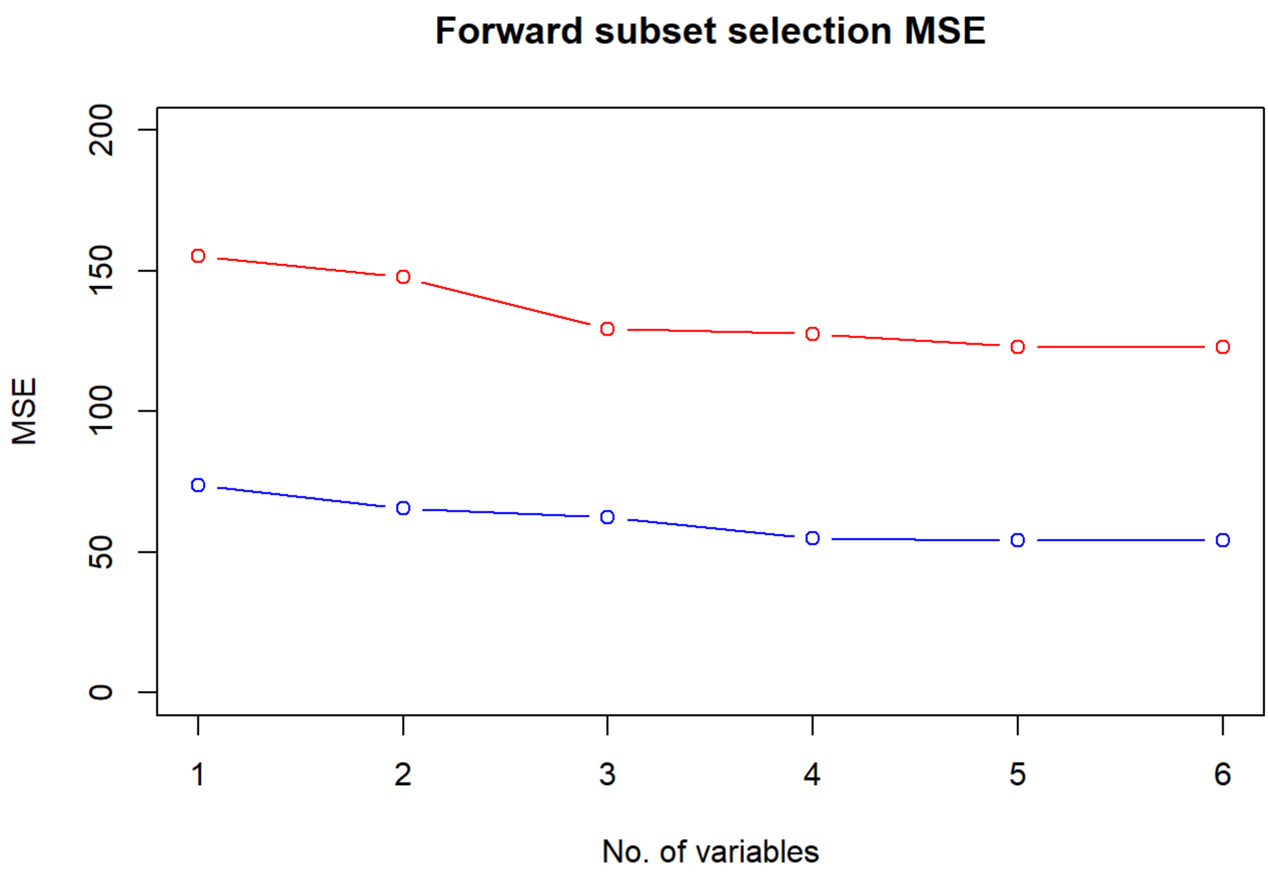



```
which.min(test_err_store)
```

```
## [1] 6
```

Minimum test error for forward subset selection using training and test data:

```
### forward subset selection:

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[,xvars]%*%coefi
}

# create objects to store error.

train_err_store1 <- matrix(rep(NA, 6))
test_err_store1 <- matrix(rep(NA, 6))
regfit.fwd <- regsubsets(PriceUnit~., data = dats, nbest = 1, nvmax = 6, method = "forward")
# perform subset selection

for (i in 1:6){
    # make the predictions
    y_hat_train1 = predict(regfit.fwd, newdata = train, id = i)
    y_hat_test1 = predict(regfit.fwd, newdata = test, id = i)

    # compare the prediction with the true
    train_err_store1[i] = (1/length(y_true_train))*sum((y_true_train-y_hat_train1)^2)
    test_err_store1[i] = (1/length(y_true_test))*sum((y_true_test-y_hat_test1)^2)

}

plot(train_err_store1, col = "blue", type = "b", xlab = "No. of variables", ylab = "MSE", yli
m = c(0,200),main="Forward subset selection MSE")
lines(test_err_store1, col = "red", type = "b")
```

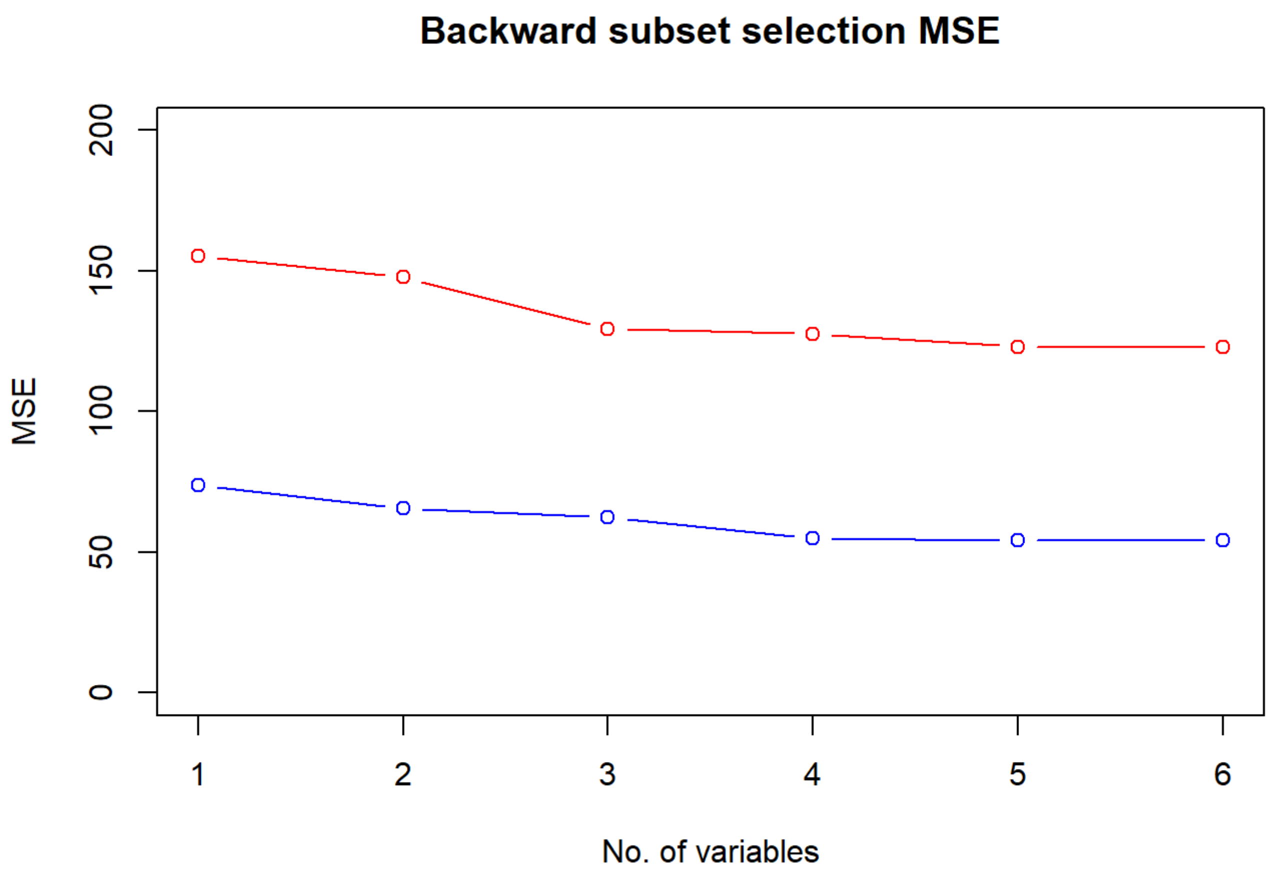



```
which.min(test_err_store1)
```

```
## [1] 6
```

Minimum test error for Backward subset selection using training and test data:

```r
### backward subset selection:

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[,xvars]%*%coefi
}

# create objects to store error.

train_err_store2 <- matrix(rep(NA, 6))
test_err_store2 <- matrix(rep(NA, 6))
regfit.bwd <- regsubsets(PriceUnit~., data = dats, nbest = 1, nvmax = 6, method = "backward")
# perform subset selection

for (i in 1:6){
    # make the predictions
    y_hat_train2 = predict(regfit.bwd, newdata = train, id = i)
    y_hat_test2 = predict(regfit.bwd, newdata = test, id = i)

    # compare the prediction with the true
    train_err_store2[i] = (1/length(y_true_train))*sum((y_true_train-y_hat_train2)^2)
    test_err_store2[i] = (1/length(y_true_test))*sum((y_true_test-y_hat_test2)^2)

}

plot(train_err_store2, col = "blue", type = "b", xlab = "No. of variables", ylab = "MSE", yli
m = c(0,200), main="Backward subset selection MSE")
lines(test_err_store2, col = "red", type = "b")
```




```
which.min(test_err_store2)
```

```
## [1] 6
```

Here i used three methods best,forward and backward subset selections, in all three methods minimum values of cp = 5, BIC = 5 and RSS = 6 are same and maximum value of adjusted R square = 5 is also same in three methods.

Next i divided the data into training and test for test errors in all three methods best,forward and backward subset selections i got the same minimum test error = 6.