STA 545 Statistical Data Mining I, Fall 2020

Homework 3

Stella Liao

September 23, 2020

1. (40 points) This problem involves the Boston data set, which we discussed in the data analysis example about subset selection. 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?

1) The simple linear regression model between 'crim' and 'zn'

```
simple.lm.fit.zn = lm(crim ~ zn, Boston)
summary(simple.lm.fit.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
```

2) The simple linear regression model between 'crim' and 'indus'

```
simple.lm.fit.indus = lm(crim ~ indus, Boston)
summary(simple.lm.fit.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
```

3) The simple linear regression model between 'crim' and 'chas'

```
simple.lm.fit.chas = lm(crim ~ chas, Boston)
summary(simple.lm.fit.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
```

4) The simple linear regression model between 'crim' and 'nox'

```
simple.lm.fit.nox = lm(crim ~ nox, Boston)
summary(simple.lm.fit.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
```

5) The simple linear regression model between 'crim' and 'rm'

```
simple.lm.fit.rm = lm(crim ~ rm, Boston)
summary(simple.lm.fit.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
```

6) The simple linear regression model between 'crim' and 'age'

```
simple.lm.fit.age = lm(crim ~ age, Boston)
summary(simple.lm.fit.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
```

7) The simple linear regression model between 'crim' and 'dis'

```r
simple.lm.fit.dis = lm(crim ~ dis, Boston)
summary(simple.lm.fit.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
```

8) The simple linear regression model between 'crim' and 'rad'

```r
simple.lm.fit.rad = lm(crim ~ rad, Boston)
summary(simple.lm.fit.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
```

9) The simple linear regression model between 'crim' and 'tax'

```r
simple.lm.fit.tax = lm(crim ~ tax, Boston)
summary(simple.lm.fit.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
```

10) The simple linear regression model between 'crim' and 'ptratio'

```
simple.lm.fit.ptratio = lm(crim ~ ptratio, Boston)
summary(simple.lm.fit.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
```

11) The simple linear regression model between 'crim' and 'black'

```
simple.lm.fit.black = lm(crim ~ black, Boston)
summary(simple.lm.fit.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
```

12) The simple linear regression model between 'crim' and 'lstat'

```
simple.lm.fit.lstat = lm(crim ~ lstat, Boston)
summary(simple.lm.fit.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
```

13) The simple linear regression model between 'crim' and 'medv'

```
simple.lm.fit.medv = lm(crim ~ medv, Boston)
summary(simple.lm.fit.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
```

Based on the p-values of each model in the chunks above, we could know that, except the predictor 'chas', other variables all have a statistically significant association with the response 'crim'.

Moreover, the p-values of the variable 'indus', 'nox', 'dis', 'rad', 'tax', 'black', 'lstat', and 'medv' are all smaller than 2.2e-16, which is very close to 0, meaning that there is a much more statistically significant association between the predictor and the response in those models.

(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 $H_0 : \beta_j = 0$

```
multiple.lm.fit = lm(crim ~., Boston)
summary(multiple.lm.fit)
```

```
##
## 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
```

Based on the p-values of each variable in the multiple regression model in the chunk above, we could know that, for the predictors 'dis', 'rad', 'medv', 'zn', 'black', 'nox' and 'lstat', we could reject the null hypothesis because their p-values are small enough. Specifically,
when $\alpha = 0.001$, for the predictors 'dis' and 'rad', we could reject the null hypothesis ;

when $\alpha = 0.01$, besides the predictors above, for the predictors 'medv', we could reject the null hypothesis;

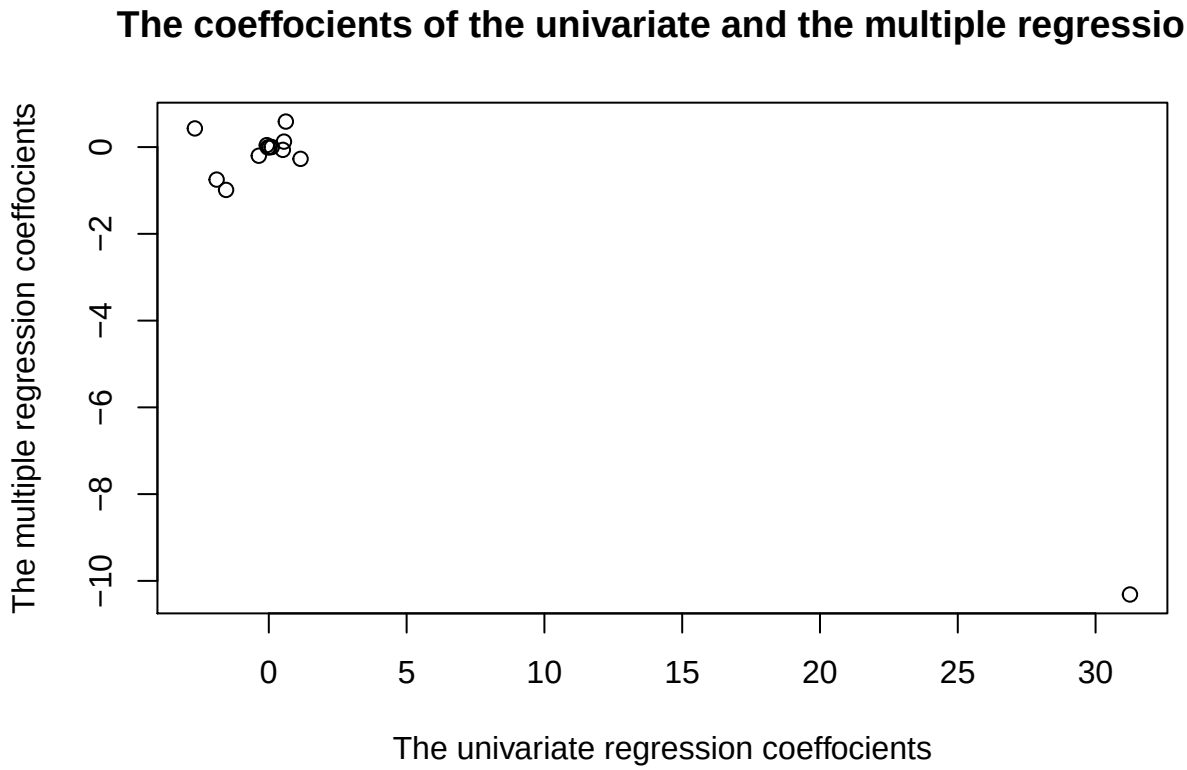
when $\alpha = 0.05$, besides the predictors above, for the predictors 'zn' and 'black', we could reject the null hypothesis;

when $\alpha = 0.1$, besides the predictors above, for the predictors 'nox' and 'lstat', we could reject the null hypothesis.

(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.

```
simple.coefs <- c()
simple.coefs <- append(simple.coefs,coef(simple.lm.fit.zn)[2])
simple.coefs <- append(simple.coefs,coef(simple.lm.fit.indus)[2])
simple.coefs <- append(simple.coefs,coef(simple.lm.fit.chas)[2])
simple.coefs <- append(simple.coefs,coef(simple.lm.fit.nox)[2])
simple.coefs <- append(simple.coefs,coef(simple.lm.fit.rm)[2])
simple.coefs <- append(simple.coefs,coef(simple.lm.fit.age)[2])
simple.coefs <- append(simple.coefs,coef(simple.lm.fit.dis)[2])
simple.coefs <- append(simple.coefs,coef(simple.lm.fit.rad)[2])
simple.coefs <- append(simple.coefs,coef(simple.lm.fit.tax)[2])
simple.coefs <- append(simple.coefs,coef(simple.lm.fit.ptratio)[2])
simple.coefs <- append(simple.coefs,coef(simple.lm.fit.black)[2])
simple.coefs <- append(simple.coefs,coef(simple.lm.fit.lstat)[2])
simple.coefs <- append(simple.coefs,coef(simple.lm.fit.medv)[2])

multiple.coefs <- coef(multiple.lm.fit)[2:14]
plot(x = simple.coefs, y = multiple.coefs,
     xlab = "The univariate regression coeffocients",
     ylab = "The multiple regression coeffocients",
     main = "The coeffocients of the univariate and the multiple regression",)
```



(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 $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon$, and test $H_0 : \beta_2 = \beta_3 = 0$.

1) The polynomial regression model between 'crim' and 'zn'

```r
poly.lm.fit.zn = lm(crim ~ zn + I(zn^2) + I(zn^3), Boston)
summary(poly.lm.fit.zn)
```

```
##
## 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
```

2) The polynomial regression model between 'crim' and 'indus'

```r
poly.lm.fit.indus = lm(crim ~ indus + I(indus^2) + I(indus^3), Boston)
summary(poly.lm.fit.indus)
```

```
##
## 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
```

3) The polynomial regression model between 'crim' and 'chas'

```r
poly.lm.fit.chas = lm(crim ~ chas + I(chas^2) + I(chas^3), Boston)
summary(poly.lm.fit.chas)
```

```
##
```

```
## Call:
## lm(formula = crim ~ chas + I(chas^2) + I(chas^3), data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.738 -3.661 -3.435  0.018 85.232
##
## Coefficients: (2 not defined because of singularities)
##             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
## I(chas^2)         NA         NA      NA       NA
## I(chas^3)         NA         NA      NA       NA
## ---
## 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
```

4) The polynomial regression model between 'crim' and 'nox'

```
poly.lm.fit.nox = lm(crim ~ nox + I(nox^2) + I(nox^3), Boston)
summary(poly.lm.fit.nox)
```

```
##
## 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
```

5) The polynomial regression model between 'crim' and 'rm'

```
poly.lm.fit.rm = lm(crim ~ rm + I(rm^2) + I(rm^3), Boston)
summary(poly.lm.fit.rm)
```

```
##
## 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
```

6) The polynomial regression model between 'crim' and 'age'

```
poly.lm.fit.age = lm(crim ~ age + I(age^2) + I(age^3), Boston)
summary(poly.lm.fit.age)
```

```
##
## 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
```

7) The polynomial regression model between 'crim' and 'dis'

```
poly.lm.fit.dis = lm(crim ~ dis + I(dis^2) + I(dis^3), Boston)
summary(poly.lm.fit.dis)
```

```
##
## 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
```

8) The polynomial regression model between 'crim' and 'rad'

```
poly.lm.fit.rad = lm(crim ~ rad + I(rad^2) + I(rad^3), Boston)
summary(poly.lm.fit.rad)
```

```
##
## 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
```

9) The polynomial regression model between 'crim' and 'tax'

```
poly.lm.fit.tax = lm(crim ~ tax + I(tax^2) + I(tax^3), Boston)
summary(poly.lm.fit.tax)
```

```
##
## 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
```

10) The polynomial regression model between 'crim' and 'ptratio'

```
poly.lm.fit.ptratio = lm(crim ~ ptratio + I(ptratio^2) + I(ptratio^3), Boston)
summary(poly.lm.fit.ptratio)
```

```
##
## 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
```

11) The polynomial regression model between 'crim' and 'black'

```
poly.lm.fit.black = lm(crim ~ black + I(black^2) + I(black^3), Boston)
summary(poly.lm.fit.black)
```

```
##
## 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
```

12) The polynomial regression model between 'crim' and 'lstat'

```
poly.lm.fit.lstat = lm(crim ~ lstat + I(lstat^2) + I(lstat^3), Boston)
summary(poly.lm.fit.lstat)
```

```
##
## 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
```

13) The polynomial regression model between 'crim' and 'medv'

```
poly.lm.fit.medv = lm(crim ~ medv + I(medv^2) + I(medv^3), Boston)
summary(poly.lm.fit.medv)
```

```
##
## 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
```

For the variables 'indus', 'nox', 'age', 'dis', 'ptratio', and 'medv', we could reject the null hypothesis that H_0: $\beta_2 = \beta_3 = 0$, because squared and cubed terms of each model of these variables are statistically significant, which means there are evidence of a non-linear relationship with those variables.

And for the remaining variables, so far, there is no evidence of a non-linear relationship between the predictor and outcome variables.

2. (30 points) We perform best subset, forward selection, and backward elimination selection on a single data set. For each approach, we obtain p+1 models, containing 0, 1, 2, . . . , p predictors. Explain your answers:

(a) Which of the three models with k predictors has the smallest training error?

The model with best subset selection has the smallest training error because it considers every possible model with k predictors.

(b) Which of the three models with k predictors has the smallest test error?

It depends. Best subset selection might have the smallest test error because it will consider more models than the other methods. However, the other methods might pick a model with smaller test error by luck.

(c) True or False:

(i) The predictors in the k-variable model identified by forward selection are a subset of the predictors in the (k+1)-variable model identified by forward selection. True

(ii) The predictors in the k-variable model identified by backward elimination are a subset of the predictors in the (k + 1)variable model identified by backward elimination. True

(iii) The predictors in the k-variable model identified by backward elimination are a subset of the predictors in the (k + 1)variable model identified by forward selection. False

(iv) The predictors in the k-variable model identified by forward selection are a subset of the predictors in the (k+1)-variable model identified by backward elimination. False

(v) The predictors in the k-variable model identified by best subset are a subset of the predictors in the (k + 1)-variable model identified by best subset selection. False

3. (30 points) In this exercise, we will generate simulated data, and will then use this data to perform best subset selection, forward selection, backward elimination. Make sure to use set.seed(1) prior to starting part (a) to ensure consistent results.

(a) Use the rnorm() function to generate a predictor X of length n = 100, as well as a noise vector of ϵ length n = 100.

```
set.seed(1)
X <- rnorm(100)
epsilon <- rnorm(100)
```

(b) Generate a response vector Y of length n = 100 according to the model $Y = 1 + 2.5X + 2X^2 + X^3 + \epsilon$.

```
Y <- 1 + 2.5 * X + 2 * X^2 + X^3 + epsilon
```

(c) Use the regsubsets() function to perform best subset selection, forward selection, and backward elimination in order to choose the best model containing the predictors X, X^2 , ..., X^{10} . Report the coefficients of the selected models.

1) best subset selection

```r
library(leaps)
data1 <- data.frame(Y, X)
regfit.full <- regsubsets(Y ~ poly(X, 10), data1, nvmax = 10, method = "exhaustive")
reg.summary <- summary(regfit.full)


which.min(reg.summary$bic)
```

```
## [1] 3
```

```r
which.min(reg.summary$cp)
```

```
## [1] 4
```

```r
which.max(reg.summary$adjr2)
```

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

2) Forward stepwise selection

```r
regfit.fwd <- regsubsets(Y ~ poly(X, 10), data1, nvmax = 10, method = "forward")
reg.summary.fwd <- summary(regfit.fwd)

which.min(reg.summary.fwd$bic)
```

```
## [1] 3
```

```r
which.min(reg.summary.fwd$cp)
```

```
## [1] 4
```

```r
which.max(reg.summary.fwd$adjr2)
```

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

1) Backward stepwise selection

```r
regfit.bwd <- regsubsets(Y ~ poly(X, 10), data1, nvmax = 10, method = "backward")
reg.summary.bwd <- summary(regfit.bwd)

which.min(reg.summary.bwd$bic)
```

```
## [1] 3
```

```r
which.min(reg.summary.bwd$cp)
```

```
## [1] 4
```

```r
which.max(reg.summary.bwd$adjr2)
```

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

According to the outputs of the chunks above, we could find that, for best subset selection, with BIC, we choose the 3-variables model; with C_p, we choose the 4-variables model, and with adjusted R^2 we choose the 5-variables model. And with forward stepwise selection and backward stepwise selection, the results are same with those when using best subset selection.

Therefore, I choose 4-variable model with best subeset selection as the best model. The coefficients are shown below.

```r
coef(regfit.full, which.min(reg.summary$cp))
```

```
##  (Intercept) poly(X, 10)1 poly(X, 10)2 poly(X, 10)3 poly(X, 10)5
##     3.066148    46.669401    24.380559    15.237294     1.480188
```