Exam2-Part2-Boston housing

Take home (65 pts) Read the question carefully and write your answers briefly supporting your conclusions with plots and statistical quantities.

You need to submit the html file and the Rmd file. Before submitting verify your html file that it includes all necessary plots and also check all necessary values are printed in the html.

Note that the questions below are open ended. There is no fixed “correct” answer. Try to use various ideas and techniques we have seen during the class.

The data set Boston contains data on Boston housing prices. The data consist of the 506 houses in Boston area.

The response variable is

- Y = medv = median value of owner-occupied homes in $1000s.

The predictor variables are:

- crim= per capita crime rate by town,
- zn= proportion of residential land zoned for lots over 25,000 sq.ft.,
- indus= proportion of non-retail business acres per town,
- chas= Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
- nox= nitrogen oxides concentration (parts per 10 million)
- rm= average number of rooms per dwelling,
- age= proportion of owner-occupied units built prior to 1940,
- dis= weighted mean of distances to five Boston employment centers
- rad= index of accessibility to radial highways
- tax = full-value property-tax rate per $10,000
- pratio= pupil-teacher ratio by town
- lstat = lower status of the population (percent). Your goal is to develop a model that predicts median value of the house (medv). You start with the multiple linear regression model using all of the 12 regressors (this is your base model). Answer the below questions. In all parts write your model clearly. In addition to writing your justification clearly, print the critical values and display the plots you use.

a. **Base model:** Fit a multiple linear regression model using all 12 regressors.

Answer goes here (model and summary):

```
library(readxl)
Boston <- read_excel("Downloads/Boston_housing.xlsx")

Boston.BM<- lm(medv~crim+zn+indus+as.factor(chas)+nox+rm+age+dis+as.factor(rad)+tax+ptra
tio+lstat, data = Boston)
summary(Boston.BM)
```

```
##
## Call:
## lm(formula = medv ~ crim + zn + indus + as.factor(chas) + nox +
##     rm + age + dis + as.factor(rad) + tax + ptratio + lstat,
##     data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.6357  -2.7013  -0.5723   1.8160  25.9979
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       40.398739   5.296438   7.628 1.27e-13 ***
## crim              -0.121816   0.032858  -3.707 0.000234 ***
## zn                 0.055525   0.014314   3.879 0.000119 ***
## indus              0.016795   0.064363   0.261 0.794250
## as.factor(chas)1   2.677692   0.872194   3.070 0.002260 **
## nox              -18.455862   3.933930  -4.691 3.53e-06 ***
## rm                 3.511231   0.423837   8.284 1.16e-15 ***
## age                0.003511   0.013353   0.263 0.792741
## dis               -1.568899   0.204235  -7.682 8.72e-14 ***
## as.factor(rad)2    1.527760   1.494794   1.022 0.307264
## as.factor(rad)3    4.698681   1.350945   3.478 0.000550 ***
## as.factor(rad)4    2.606331   1.201262   2.170 0.030516 *
## as.factor(rad)5    2.864862   1.221675   2.345 0.019427 *
## as.factor(rad)6    1.283888   1.480915   0.867 0.386394
## as.factor(rad)7    4.917263   1.589585   3.093 0.002093 **
## as.factor(rad)8    4.820869   1.509140   3.194 0.001492 **
## as.factor(rad)24   7.123585   1.807059   3.942 9.26e-05 ***
## tax               -0.009111   0.003939  -2.313 0.021146 *
## ptratio           -0.960781   0.146134  -6.575 1.26e-10 ***
## lstat             -0.557596   0.050584 -11.023  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.749 on 486 degrees of freedom
## Multiple R-squared:  0.7434, Adjusted R-squared:  0.7334
## F-statistic: 74.12 on 19 and 486 DF,  p-value: < 2.2e-16
```

```
attach(Boston)
```

b. **Interaction Terms:** Give a model that uses base model and includes interaction terms Crim x age, rm x tax, rm x ptratio, tax x ptratio, nox x crim, nox xage and 3 additional interaction terms of your choice. Check if any of these interaction terms contribute to the model. Do backwards selection to create a simpler model with interactions.

Answer goes here (model and summary):

```
Boston.IT<- lm(medv~crim+zn+indus+as.factor(chas)+nox+rm+age+dis+as.factor(rad)+tax+ptra
tio+lstat+crim*age+rm*tax+rm*ptratio+tax*ptratio+nox*crim+nox*age+indus*tax+crim*tax+ptr
atio*crim, data = Boston)
summary(Boston.IT)
```

```
##
## Call:
## lm(formula = medv ~ crim + zn + indus + as.factor(chas) + nox +
##     rm + age + dis + as.factor(rad) + tax + ptratio + lstat +
##     crim * age + rm * tax + rm * ptratio + tax * ptratio + nox *
##     crim + nox * age + indus * tax + crim * tax + ptratio * crim,
##     data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.9424  -2.2374  -0.3788   1.3927  26.8693
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -9.635e+01  2.268e+01  -4.247 2.60e-05 ***
## crim              5.849e+00  2.807e+00   2.084 0.037679 *
## zn                2.534e-02  1.396e-02   1.816 0.070027 .
## indus             2.188e-01  1.332e-01   1.643 0.101125
## as.factor(chas)1  3.571e+00  7.586e-01   4.707 3.31e-06 ***
## nox               6.504e-02  1.296e+01   0.005 0.995998
## rm                2.274e+01  2.406e+00   9.452  < 2e-16 ***
## age               4.881e-02  6.902e-02   0.707 0.479760
## dis              -8.554e-01  1.904e-01  -4.494 8.79e-06 ***
## as.factor(rad)2   1.561e+00  1.310e+00   1.192 0.234000
## as.factor(rad)3   4.707e+00  1.187e+00   3.964 8.49e-05 ***
## as.factor(rad)4   2.016e+00  1.073e+00   1.879 0.060817 .
## as.factor(rad)5   2.499e+00  1.072e+00   2.330 0.020218 *
## as.factor(rad)6   1.912e+00  1.308e+00   1.462 0.144408
## as.factor(rad)7   3.874e+00  1.383e+00   2.801 0.005300 **
## as.factor(rad)8   3.272e+00  1.329e+00   2.461 0.014196 *
## as.factor(rad)24  6.252e+00  1.778e+00   3.516 0.000479 ***
## tax               6.487e-02  3.815e-02   1.700 0.089715 .
## ptratio           3.543e+00  1.305e+00   2.714 0.006883 **
## lstat            -5.098e-01  4.603e-02 -11.075  < 2e-16 ***
## crim:age          6.644e-03  3.281e-03   2.025 0.043413 *
## rm:tax           -1.429e-02  2.234e-03  -6.395 3.82e-10 ***
## rm:ptratio       -6.943e-01  1.557e-01  -4.459 1.03e-05 ***
## tax:ptratio       1.061e-03  1.986e-03   0.534 0.593387
## crim:nox         -1.303e+00  6.620e-01  -1.968 0.049611 *
## nox:age          -1.398e-01  1.420e-01  -0.984 0.325547
## indus:tax        -3.664e-04  3.594e-04  -1.020 0.308475
## crim:tax         -1.922e-03  2.929e-03  -0.656 0.511955
## crim:ptratio     -2.219e-01  1.916e-01  -1.158 0.247438
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.098 on 477 degrees of freedom
## Multiple R-squared:  0.8125, Adjusted R-squared:  0.8015
## F-statistic: 73.82 on 28 and 477 DF,  p-value: < 2.2e-16
```

```
Boston.IT2= update(Boston.IT,~.- nox)
summary(Boston.IT2)
```

```
##
## Call:
## lm(formula = medv ~ crim + zn + indus + as.factor(chas) + rm +
##     age + dis + as.factor(rad) + tax + ptratio + lstat + crim:age +
##     rm:tax + rm:ptratio + tax:ptratio + crim:nox + nox:age +
##     indus:tax + crim:tax + crim:ptratio, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.9395  -2.2382  -0.3797   1.3924  26.8683
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -9.631e+01  2.183e+01  -4.413 1.26e-05 ***
## crim              5.850e+00  2.800e+00   2.090 0.037177 *
## zn                2.531e-02  1.295e-02   1.955 0.051217 .
## indus             2.187e-01  1.323e-01   1.653 0.099088 .
## as.factor(chas)1  3.571e+00  7.573e-01   4.715 3.18e-06 ***
## rm                2.274e+01  2.401e+00   9.471  < 2e-16 ***
## age               4.850e-02  3.003e-02   1.615 0.106989
## dis              -8.557e-01  1.830e-01  -4.676 3.82e-06 ***
## as.factor(rad)2   1.561e+00  1.308e+00   1.193 0.233491
## as.factor(rad)3   4.707e+00  1.183e+00   3.980 7.98e-05 ***
## as.factor(rad)4   2.016e+00  1.071e+00   1.882 0.060430 .
## as.factor(rad)5   2.500e+00  1.066e+00   2.344 0.019496 *
## as.factor(rad)6   1.913e+00  1.295e+00   1.477 0.140467
## as.factor(rad)7   3.874e+00  1.381e+00   2.804 0.005250 **
## as.factor(rad)8   3.272e+00  1.325e+00   2.470 0.013862 *
## as.factor(rad)24  6.253e+00  1.767e+00   3.539 0.000441 ***
## tax               6.485e-02  3.795e-02   1.709 0.088139 .
## ptratio           3.543e+00  1.302e+00   2.722 0.006720 **
## lstat            -5.098e-01  4.588e-02 -11.113  < 2e-16 ***
## crim:age          6.637e-03  2.964e-03   2.239 0.025585 *
## rm:tax           -1.428e-02  2.217e-03  -6.442 2.89e-10 ***
## rm:ptratio       -6.943e-01  1.550e-01  -4.481 9.31e-06 ***
## tax:ptratio       1.062e-03  1.983e-03   0.535 0.592668
## crim:nox         -1.303e+00  6.609e-01  -1.971 0.049253 *
## age:nox          -1.391e-01  5.127e-02  -2.714 0.006898 **
## indus:tax        -3.663e-04  3.584e-04  -1.022 0.307246
## crim:tax         -1.924e-03  2.917e-03  -0.660 0.509869
## crim:ptratio     -2.218e-01  1.913e-01  -1.160 0.246778
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.093 on 478 degrees of freedom
## Multiple R-squared:  0.8125, Adjusted R-squared:  0.8019
## F-statistic: 76.72 on 27 and 478 DF,  p-value: < 2.2e-16
```

```
Boston.IT3= update(Boston.IT2,~.-tax*ptratio)
summary(Boston.IT3)
```

```
##
## Call:
## lm(formula = medv ~ crim + zn + indus + as.factor(chas) + rm +
##     age + dis + as.factor(rad) + lstat + crim:age + rm:tax +
##     rm:ptratio + crim:nox + age:nox + indus:tax + crim:tax +
##     crim:ptratio, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.590  -2.571  -0.478   1.790  25.679
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       4.3100153  3.4575740   1.247 0.213172
## crim              0.0276931  2.5699636   0.011 0.991407
## zn                0.0474615  0.0137271   3.458 0.000594 ***
## indus            -0.2743951  0.1117556  -2.455 0.014429 *
## as.factor(chas)1  2.8210865  0.8151229   3.461 0.000586 ***
## rm                8.8659180  0.6052396  14.649  < 2e-16 ***
## age               0.0607075  0.0323404   1.877 0.061104 .
## dis              -1.1493713  0.1943650  -5.913 6.36e-09 ***
## as.factor(rad)2   1.8944467  1.4138169   1.340 0.180894
## as.factor(rad)3   3.8763580  1.2770792   3.035 0.002533 **
## as.factor(rad)4   2.1460206  1.1580557   1.853 0.064477 .
## as.factor(rad)5   2.8963317  1.1513826   2.516 0.012210 *
## as.factor(rad)6   2.6115996  1.4009698   1.864 0.062911 .
## as.factor(rad)7   4.5657020  1.4913667   3.061 0.002326 **
## as.factor(rad)8   4.4139516  1.4223641   3.103 0.002027 **
## as.factor(rad)24  9.1352099  1.7170950   5.320 1.59e-07 ***
## lstat            -0.5319757  0.0487698 -10.908  < 2e-16 ***
## crim:age          0.0084197  0.0031944   2.636 0.008665 **
## rm:tax           -0.0055165  0.0008270  -6.670 7.04e-11 ***
## rm:ptratio       -0.1592776  0.0249827  -6.376 4.28e-10 ***
## crim:nox         -1.3107301  0.7154998  -1.832 0.067583 .
## age:nox          -0.1580175  0.0552165  -2.862 0.004396 **
## indus:tax         0.0011841  0.0002703   4.380 1.46e-05 ***
## crim:tax         -0.0054103  0.0030043  -1.801 0.072354 .
## crim:ptratio      0.1741724  0.1829360   0.952 0.341527
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.435 on 481 degrees of freedom
## Multiple R-squared:  0.7785, Adjusted R-squared:  0.7674
## F-statistic: 70.44 on 24 and 481 DF,  p-value: < 2.2e-16
```

```
Boston.ITF= update(Boston.IT3,~.-crim)
summary(Boston.ITF)
```

```
##
## Call:
## lm(formula = medv ~ zn + indus + as.factor(chas) + rm + age +
##     dis + as.factor(rad) + lstat + crim:age + rm:tax + rm:ptratio +
##     crim:nox + age:nox + indus:tax + crim:tax + crim:ptratio,
##     data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.5895  -2.5698  -0.4786   1.7872  25.6796
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       4.3122447  3.4477964   1.251 0.211642
## zn                0.0474543  0.0136970   3.465 0.000578 ***
## indus            -0.2741699  0.1096703  -2.500 0.012752 *
## as.factor(chas)1  2.8213666  0.8138626   3.467 0.000574 ***
## rm                8.8677469  0.5803503  15.280  < 2e-16 ***
## age               0.0606034  0.0308342   1.965 0.049934 *
## dis              -1.1492385  0.1937725  -5.931 5.75e-09 ***
## as.factor(rad)2   1.8937441  1.4108470   1.342 0.180139
## as.factor(rad)3   3.8776503  1.2701162   3.053 0.002391 **
## as.factor(rad)4   2.1447319  1.1506684   1.864 0.062943 .
## as.factor(rad)5   2.8975848  1.1443052   2.532 0.011652 *
## as.factor(rad)6   2.6110856  1.3987045   1.867 0.062538 .
## as.factor(rad)7   4.5666931  1.4869831   3.071 0.002253 **
## as.factor(rad)8   4.4148538  1.4184242   3.113 0.001965 **
## as.factor(rad)24  9.1322636  1.6934265   5.393 1.09e-07 ***
## lstat            -0.5320396  0.0483578 -11.002  < 2e-16 ***
## age:crim          0.0084210  0.0031889   2.641 0.008542 **
## rm:tax           -0.0055158  0.0008238  -6.696 5.99e-11 ***
## rm:ptratio       -0.1594187  0.0212591  -7.499 3.12e-13 ***
## crim:nox         -1.3102825  0.7135518  -1.836 0.066932 .
## age:nox          -0.1578292  0.0523218  -3.017 0.002692 **
## indus:tax         0.0011840  0.0002699   4.387 1.41e-05 ***
## crim:tax         -0.0054212  0.0028269  -1.918 0.055737 .
## crim:ptratio      0.1758814  0.0910785   1.931 0.054057 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.431 on 482 degrees of freedom
## Multiple R-squared:  0.7785, Adjusted R-squared:  0.7679
## F-statistic: 73.65 on 23 and 482 DF,  p-value: < 2.2e-16
```

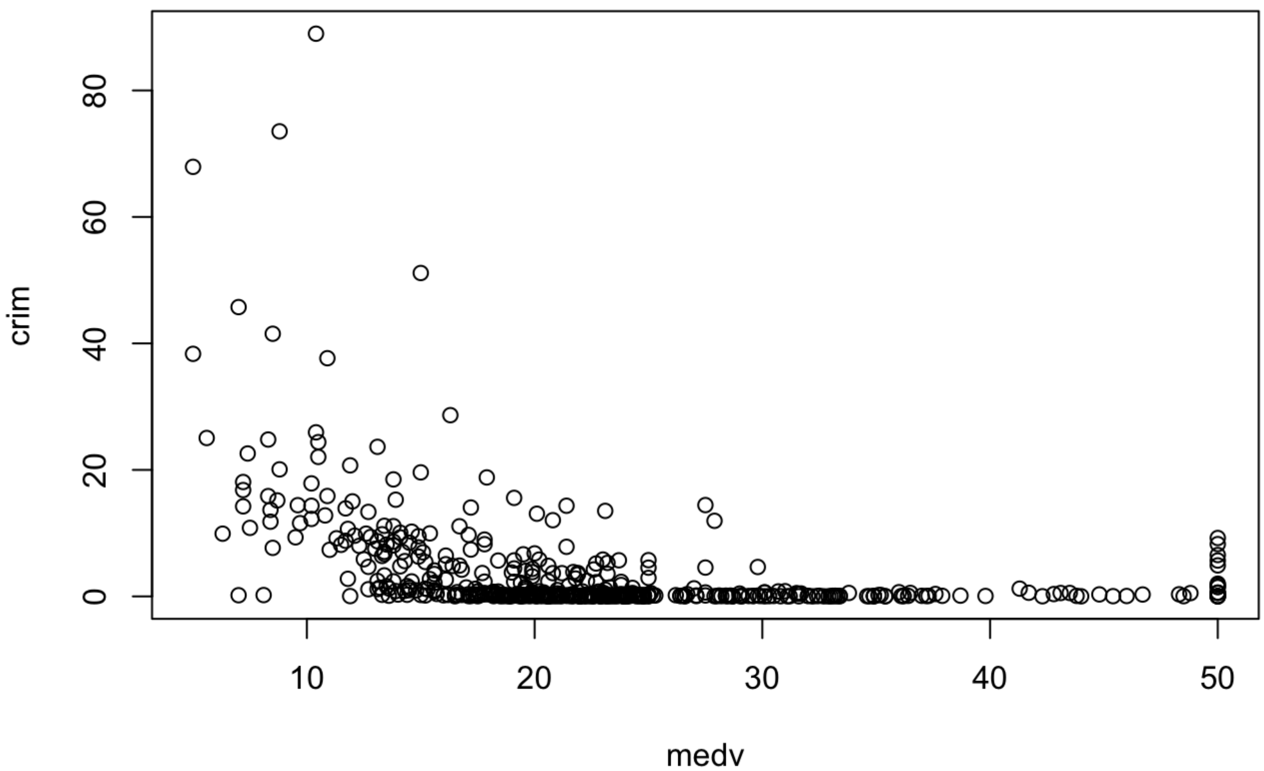
c. **Transformation of variables:** Try various transformations on the base model, then propose a transformation (on prediction variable medv or on regressors) that you think it might be helpful to linearize the model (or to improve it). Then fit a model using this transformation. Explain which variables were transformed and why.

Answer goes here (model, summary and explanation):

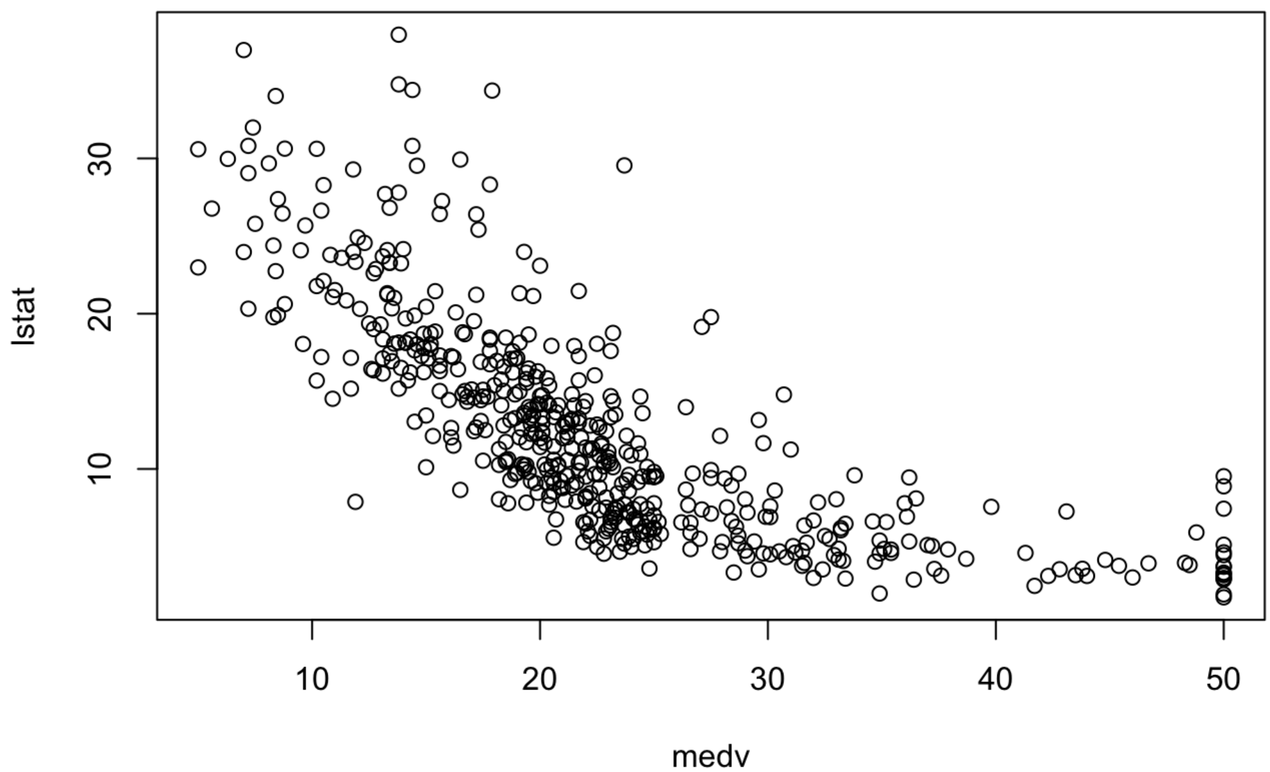
```
library(car)
```

```
## Loading required package: carData
```

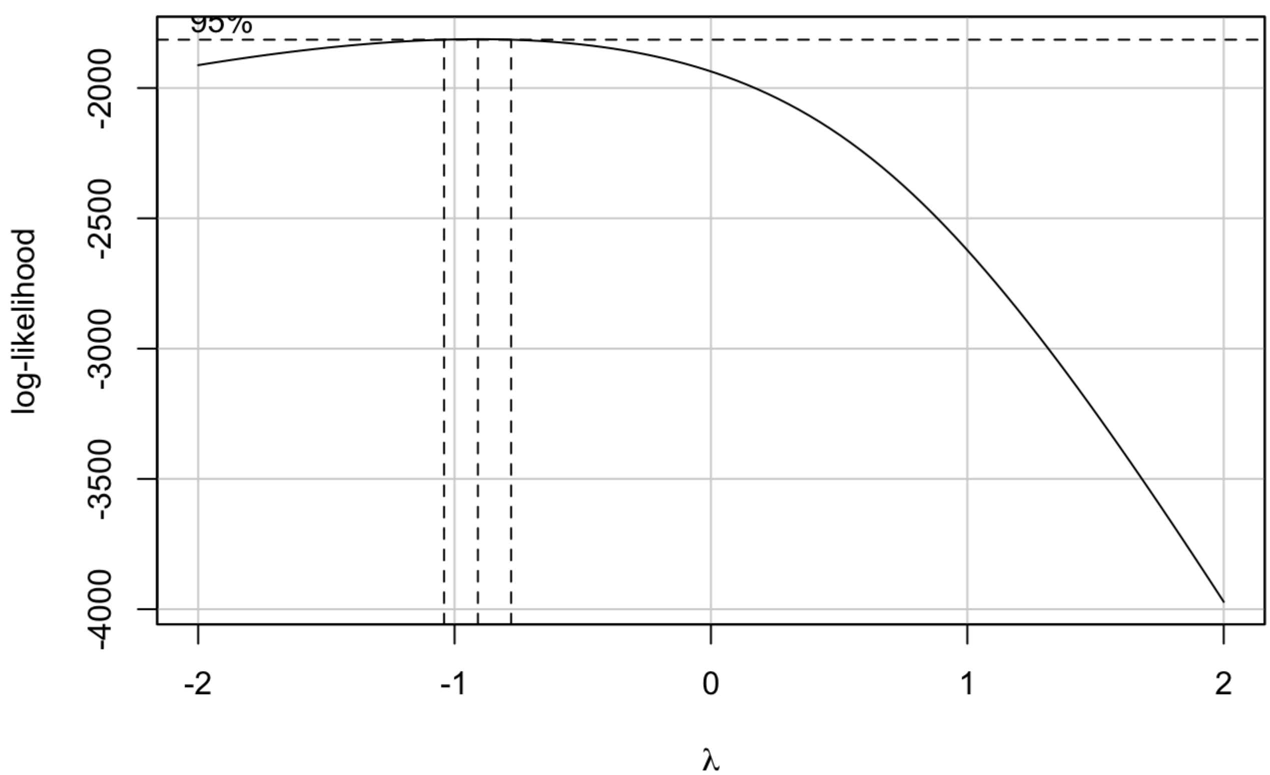
```
plot(crim~medv)
```



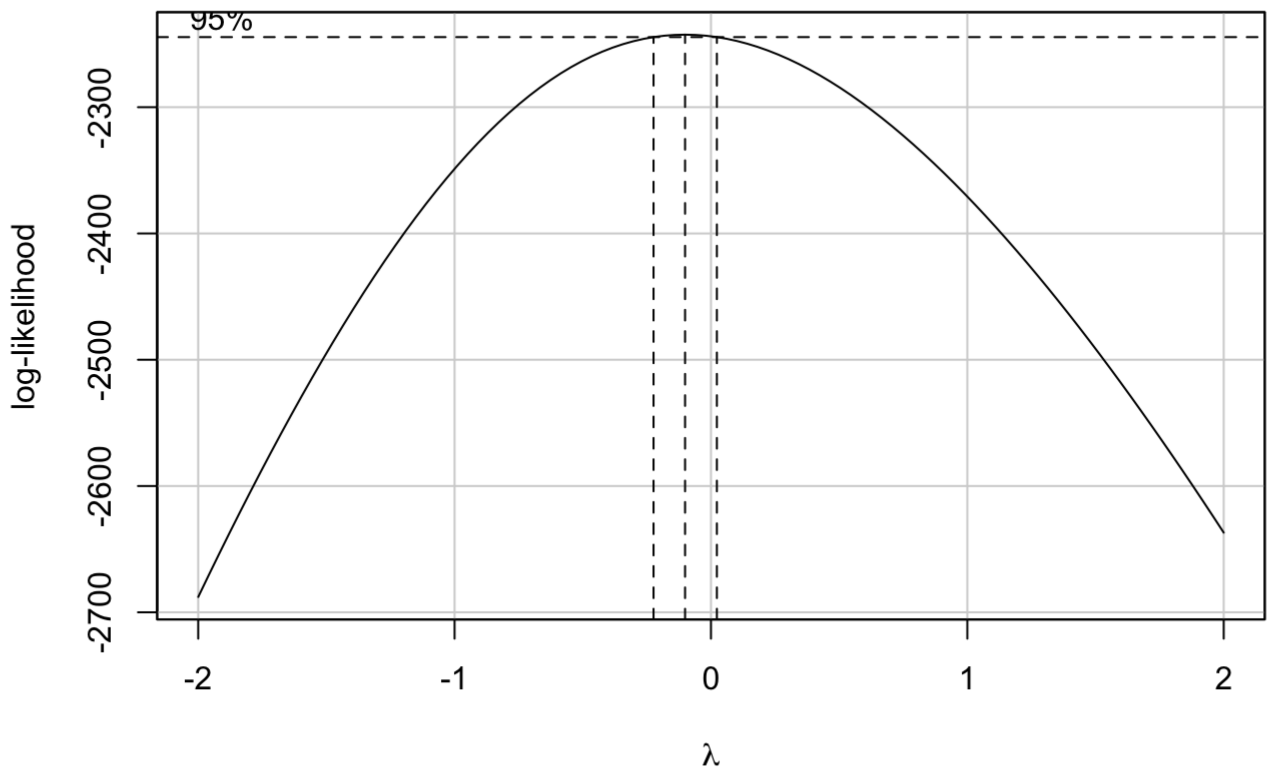

```
plot(lstat~medv)
```




```
boxCox(lm(crim~medv), family= "yjPower")
```




```
boxCox(lm(lstat~medv), family= "yjPower")
```




```
crimT<- yjPower(crim, -.9)
lstatT<- yjPower(lstat, -.1)
Boston.TM<- lm(medv~crimT+zn+indus+as.factor(chas)+nox+rm+age+dis+as.factor(rad)+tax+ptr
atio+lstatT, data = Boston)
summary(Boston.TM)
```

```
##
## Call:
## lm(formula = medv ~ crimT + zn + indus + as.factor(chas) + nox +
##     rm + age + dis + as.factor(rad) + tax + ptratio + lstatT,
##     data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.0180  -2.5715  -0.2805   1.9719  25.5426
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       64.316215   5.319075  12.092  < 2e-16 ***
## crimT             -0.111819   1.708098  -0.065 0.947832
## zn                 0.025549   0.013179   1.939 0.053136 .
## indus              0.027352   0.059198   0.462 0.644258
## as.factor(chas)1   2.418201   0.802432   3.014 0.002717 **
## nox              -15.645542   3.935633  -3.975 8.09e-05 ***
## rm                 2.345261   0.404626   5.796 1.22e-08 ***
## age                0.029704   0.012687   2.341 0.019618 *
## dis               -1.184617   0.187578  -6.315 6.09e-10 ***
## as.factor(rad)2    1.445469   1.372572   1.053 0.292814
## as.factor(rad)3    4.214801   1.242199   3.393 0.000748 ***
## as.factor(rad)4    2.579657   1.126363   2.290 0.022434 *
## as.factor(rad)5    2.569778   1.129175   2.276 0.023292 *
## as.factor(rad)6    2.408636   1.362104   1.768 0.077635 .
## as.factor(rad)7    4.601021   1.473805   3.122 0.001904 **
## as.factor(rad)8    3.878574   1.416535   2.738 0.006407 **
## as.factor(rad)24   5.749277   1.984332   2.897 0.003933 **
## tax               -0.009116   0.003617  -2.520 0.012043 *
## ptratio           -0.858215   0.137905  -6.223 1.05e-09 ***
## lstatT           -13.767180   0.833010 -16.527  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.36 on 486 degrees of freedom
## Multiple R-squared:  0.7837, Adjusted R-squared:  0.7752
## F-statistic: 92.67 on 19 and 486 DF,  p-value: < 2.2e-16
```

```
#I will try another transformation this time only transforming the response variable
Boston.TM2<- lm(log(medv)~crim+zn+indus+as.factor(chas)+nox+rm+age+dis+as.factor(rad)+ta
x+ptratio+lstat, data = Boston)
summary(Boston.TM2)
```

```
##
## Call:
## lm(formula = log(medv) ~ crim + zn + indus + as.factor(chas) +
##     nox + rm + age + dis + as.factor(rad) + tax + ptratio + lstat,
##     data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68178 -0.10160 -0.01198  0.09992  0.81278
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       4.2347902  0.2138514  19.802  < 2e-16 ***
## crim             -0.0108009  0.0013267  -8.141 3.32e-15 ***
## zn                0.0015374  0.0005779   2.660  0.00807 **
## indus             0.0024672  0.0025988   0.949  0.34290
## as.factor(chas)1  0.1070532  0.0352161   3.040  0.00249 **
## nox              -0.8118549  0.1588382  -5.111 4.61e-07 ***
## rm                0.0800587  0.0171130   4.678 3.76e-06 ***
## age               0.0003328  0.0005392   0.617  0.53734
## dis              -0.0517614  0.0082463  -6.277 7.66e-10 ***
## as.factor(rad)2   0.0850869  0.0603545   1.410  0.15924
## as.factor(rad)3   0.1774313  0.0545464   3.253  0.00122 **
## as.factor(rad)4   0.1015640  0.0485027   2.094  0.03678 *
## as.factor(rad)5   0.1321144  0.0493269   2.678  0.00765 **
## as.factor(rad)6   0.1004904  0.0597941   1.681  0.09348 .
## as.factor(rad)7   0.2092091  0.0641818   3.260  0.00119 **
## as.factor(rad)8   0.1931184  0.0609337   3.169  0.00162 **
## as.factor(rad)24  0.3355303  0.0729627   4.599 5.43e-06 ***
## tax              -0.0005212  0.0001590  -3.277  0.00112 **
## ptratio          -0.0364930  0.0059004  -6.185 1.32e-09 ***
## lstat            -0.0304173  0.0020424 -14.893  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1917 on 486 degrees of freedom
## Multiple R-squared:  0.7883, Adjusted R-squared:   0.78
## F-statistic: 95.22 on 19 and 486 DF,  p-value: < 2.2e-16
```

```
#What if we only transform one predictor variable
Boston.TM3<- lm(medv~crim+zn+indus+as.factor(chas)+nox+rm+age+dis+as.factor(rad)+tax+ptr
atio+lstatT, data = Boston)
summary(Boston.TM3)
```

```
##
## Call:
## lm(formula = medv ~ crim + zn + indus + as.factor(chas) + nox +
##     rm + age + dis + as.factor(rad) + tax + ptratio + lstatT,
##     data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.4979  -2.5389  -0.2708   1.8781  25.0625
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       65.601308   5.203532  12.607  < 2e-16 ***
## crim              -0.140215   0.029157  -4.809 2.03e-06 ***
## zn                 0.030528   0.012885   2.369 0.018210 *
## indus              0.017518   0.057689   0.304 0.761519
## as.factor(chas)1   2.197908   0.783715   2.804 0.005242 **
## nox              -16.553602   3.531636  -4.687 3.60e-06 ***
## rm                 2.249945   0.395187   5.693 2.16e-08 ***
## age                0.030719   0.012212   2.515 0.012209 *
## dis               -1.289352   0.184539  -6.987 9.30e-12 ***
## as.factor(rad)2    1.327724   1.340175   0.991 0.322321
## as.factor(rad)3    4.194652   1.210190   3.466 0.000575 ***
## as.factor(rad)4    2.545643   1.076744   2.364 0.018461 *
## as.factor(rad)5    2.558322   1.096226   2.334 0.020015 *
## as.factor(rad)6    2.231204   1.331308   1.676 0.094391 .
## as.factor(rad)7    4.715777   1.424782   3.310 0.001003 **
## as.factor(rad)8    3.994674   1.353551   2.951 0.003318 **
## as.factor(rad)24   7.325010   1.620564   4.520 7.77e-06 ***
## tax               -0.009265   0.003534  -2.622 0.009020 **
## ptratio           -0.882820   0.131163  -6.731 4.77e-11 ***
## lstatT           -13.402069   0.817409 -16.396  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.26 on 486 degrees of freedom
## Multiple R-squared:  0.7935, Adjusted R-squared:  0.7854
## F-statistic:  98.3 on 19 and 486 DF,  p-value: < 2.2e-16
```

```
#Notice we have achived a higher R2 when only lstat is transformed. This transformation
 might be helpful to linearize the model
bsel1=step(Boston.TM3)
```

```
## Start:  AIC=1486.3
## medv ~ crim + zn + indus + as.factor(chas) + nox + rm + age +
##     dis + as.factor(rad) + tax + ptratio + lstatT
##
##                    Df Sum of Sq     RSS    AIC
## - indus             1       1.7  8822.2 1484.4
## <none>                           8820.5 1486.3
## - zn                1     101.9  8922.4 1490.1
## - age               1     114.8  8935.3 1490.8
## - tax               1     124.8  8945.3 1491.4
## - as.factor(chas)  1     142.7  8963.2 1492.4
## - nox               1     398.7  9219.2 1506.7
## - as.factor(rad)   8     669.7  9490.2 1507.3
## - crim              1     419.7  9240.2 1507.8
## - rm                1     588.3  9408.8 1517.0
## - ptratio           1     822.2  9642.7 1529.4
## - dis               1     886.0  9706.5 1532.7
## - lstatT            1    4878.9 13699.4 1707.1
##
## Step:  AIC=1484.39
## medv ~ crim + zn + as.factor(chas) + nox + rm + age + dis + as.factor(rad) +
##     tax + ptratio + lstatT
##
##                    Df Sum of Sq     RSS    AIC
## <none>                           8822.2 1484.4
## - zn                1     100.2  8922.4 1488.1
## - age               1     114.8  8937.0 1488.9
## - tax               1     137.8  8959.9 1490.2
## - as.factor(chas)  1     147.5  8969.7 1490.8
## - nox               1     417.3  9239.5 1505.8
## - crim              1     422.0  9244.2 1506.0
## - as.factor(rad)   8     692.0  9514.2 1506.6
## - rm                1     588.0  9410.1 1515.0
## - ptratio           1     820.5  9642.7 1527.4
## - dis               1     936.9  9759.1 1533.5
## - lstatT            1    4882.1 13704.3 1705.3
```

```
library(MASS)
```

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

```
## The following object is masked _by_ '.GlobalEnv':
##
##     Boston
```

```
step1 <- stepAIC(Boston.TM3, direction="both")
```

```
## Start:  AIC=1486.3
## medv ~ crim + zn + indus + as.factor(chas) + nox + rm + age +
##     dis + as.factor(rad) + tax + ptratio + lstatT
##
##                   Df Sum of Sq     RSS    AIC
## - indus            1       1.7  8822.2 1484.4
## <none>                          8820.5 1486.3
## - zn               1     101.9  8922.4 1490.1
## - age              1     114.8  8935.3 1490.8
## - tax              1     124.8  8945.3 1491.4
## - as.factor(chas)  1     142.7  8963.2 1492.4
## - nox              1     398.7  9219.2 1506.7
## - as.factor(rad)   8     669.7  9490.2 1507.3
## - crim             1     419.7  9240.2 1507.8
## - rm               1     588.3  9408.8 1517.0
## - ptratio          1     822.2  9642.7 1529.4
## - dis              1     886.0  9706.5 1532.7
## - lstatT           1    4878.9 13699.4 1707.1
##
## Step:  AIC=1484.39
## medv ~ crim + zn + as.factor(chas) + nox + rm + age + dis + as.factor(rad) +
##     tax + ptratio + lstatT
##
##                   Df Sum of Sq     RSS    AIC
## <none>                          8822.2 1484.4
## + indus            1       1.7  8820.5 1486.3
## - zn               1     100.2  8922.4 1488.1
## - age              1     114.8  8937.0 1488.9
## - tax              1     137.8  8959.9 1490.2
## - as.factor(chas)  1     147.5  8969.7 1490.8
## - nox              1     417.3  9239.5 1505.8
## - crim             1     422.0  9244.2 1506.0
## - as.factor(rad)   8     692.0  9514.2 1506.6
## - rm               1     588.0  9410.1 1515.0
## - ptratio          1     820.5  9642.7 1527.4
## - dis              1     936.9  9759.1 1533.5
## - lstatT           1    4882.1 13704.3 1705.3
```

```
step1$anova # display results
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## medv ~ crim + zn + indus + as.factor(chas) + nox + rm + age +
##     dis + as.factor(rad) + tax + ptratio + lstatT
##
## Final Model:
## medv ~ crim + zn + as.factor(chas) + nox + rm + age + dis + as.factor(rad) +
##     tax + ptratio + lstatT
##
##
##      Step Df Deviance Resid. Df Resid. Dev      AIC
## 1                           486   8820.493 1486.298
## 2 - indus  1  1.67349       487   8822.167 1484.394
```

```
#Notice that the backwards stepping and stepwise stepping both found the same model.
BestTMM<- lm(medv ~ crim + zn + as.factor(chas) + nox + rm + age + dis + as.factor(rad)
 +
              tax + ptratio + lstatT, data = Boston)
```

I choose to transform Crim and Lstat because when I looked at the relation of them to medv they both were non linear.

d. **Polynomial terms:** Eliminate 6 of the regressors from the base model, that (you think) are the least significant ones. (You can do a subjective choice, considering the nature of the data, as long as you support it. For example you can make a few joint significance test to support your choice). Now using the remaining 6 regressors propose a polynomial model that includes quadratic terms and interaction terms. Then fit this model. Answer goes here (model, summary and explanation):

```
bsel2<- step(Boston.BM)
```

```
## Start:  AIC=1596.16
## medv ~ crim + zn + indus + as.factor(chas) + nox + rm + age +
##     dis + as.factor(rad) + tax + ptratio + lstat
##
##                   Df Sum of Sq   RSS    AIC
## - indus            1      1.54 10961 1594.2
## - age              1      1.56 10961 1594.2
## <none>                         10959 1596.2
## - tax              1    120.63 11080 1599.7
## - as.factor(chas)  1    212.54 11172 1603.9
## - crim             1    309.94 11269 1608.3
## - zn               1    339.33 11299 1609.6
## - nox              1    496.32 11456 1616.6
## - as.factor(rad)   8    820.77 11780 1616.7
## - ptratio          1    974.75 11934 1637.3
## - dis              1   1330.69 12290 1652.1
## - rm               1   1547.64 12507 1661.0
## - lstat            1   2740.02 13699 1707.1
##
## Step:  AIC=1594.23
## medv ~ crim + zn + as.factor(chas) + nox + rm + age + dis + as.factor(rad) +
##     tax + ptratio + lstat
##
##                   Df Sum of Sq   RSS    AIC
## - age              1      1.55 10962 1592.3
## <none>                         10961 1594.2
## - tax              1    133.52 11094 1598.4
## - as.factor(chas)  1    218.62 11180 1602.2
## - crim             1    312.24 11273 1606.4
## - zn               1    340.09 11301 1607.7
## - as.factor(rad)   8    843.21 11804 1615.7
## - nox              1    521.81 11483 1615.8
## - ptratio          1    973.36 11934 1635.3
## - dis              1   1400.69 12362 1653.1
## - rm               1   1553.55 12514 1659.3
## - lstat            1   2743.42 13704 1705.3
##
## Step:  AIC=1592.3
## medv ~ crim + zn + as.factor(chas) + nox + rm + dis + as.factor(rad) +
##     tax + ptratio + lstat
##
##                   Df Sum of Sq   RSS    AIC
## <none>                         10962 1592.3
## - tax              1    132.35 11095 1596.4
## - as.factor(chas)  1    221.02 11184 1600.4
## - crim             1    312.36 11275 1604.5
## - zn               1    338.89 11301 1605.7
## - as.factor(rad)   8    841.91 11804 1613.7
## - nox              1    543.93 11506 1614.8
## - ptratio          1    975.77 11938 1633.5
## - dis              1   1563.90 12526 1657.8
## - rm               1   1623.80 12586 1660.2
## - lstat            1   3050.72 14013 1714.5
```

```
summary(bsel2)
```

```
##
## Call:
## lm(formula = medv ~ crim + zn + as.factor(chas) + nox + rm +
##     dis + as.factor(rad) + tax + ptratio + lstat, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.6779  -2.7212  -0.4832   1.7849  26.1280
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       40.237252   5.267117   7.639 1.16e-13 ***
## crim              -0.122189   0.032768  -3.729 0.000215 ***
## zn                 0.054603   0.014058   3.884 0.000117 ***
## as.factor(chas)1   2.712564   0.864787   3.137 0.001812 **
## nox              -17.911598   3.640033  -4.921 1.18e-06 ***
## rm                 3.519984   0.414017   8.502 2.29e-16 ***
## dis               -1.594663   0.191121  -8.344 7.44e-16 ***
## as.factor(rad)2    1.597403   1.476432   1.082 0.279816
## as.factor(rad)3    4.674248   1.346722   3.471 0.000565 ***
## as.factor(rad)4    2.616563   1.195405   2.189 0.029081 *
## as.factor(rad)5    2.852057   1.218321   2.341 0.019635 *
## as.factor(rad)6    1.222933   1.467568   0.833 0.405080
## as.factor(rad)7    4.897766   1.585188   3.090 0.002118 **
## as.factor(rad)8    4.805755   1.498994   3.206 0.001434 **
## as.factor(rad)24   6.997177   1.767284   3.959 8.64e-05 ***
## tax               -0.008623   0.003552  -2.427 0.015572 *
## ptratio           -0.954861   0.144881  -6.591 1.14e-10 ***
## lstat             -0.552240   0.047388 -11.654  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.74 on 488 degrees of freedom
## Multiple R-squared:  0.7434, Adjusted R-squared:  0.7344
## F-statistic: 83.15 on 17 and 488 DF,  p-value: < 2.2e-16
```

```
#I am going to remove age and indus from our base model since from backwards selection we notice that it is not significant
# I will also remove chas, dis, and tax as I do not belive that they would infulence the medv
Boston.P4Update<- update(bsel2,~.-as.factor(chas)-tax-dis)
summary(Boston.P4Update)
```

```
##
## Call:
## lm(formula = medv ~ crim + zn + nox + rm + as.factor(rad) + ptratio +
##     lstat, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.2080  -3.1390  -0.7746   1.9087  28.9452
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      21.511053   5.160991   4.168 3.63e-05 ***
## crim             -0.096564   0.035215  -2.742 0.006327 **
## zn               -0.007202   0.013196  -0.546 0.585453
## nox              -3.983067   3.323917  -1.198 0.231376
## rm                4.292130   0.438902   9.779  < 2e-16 ***
## as.factor(rad)2   3.144138   1.580501   1.989 0.047219 *
## as.factor(rad)3   5.290236   1.450650   3.647 0.000294 ***
## as.factor(rad)4   3.166493   1.285905   2.462 0.014141 *
## as.factor(rad)5   2.980062   1.315260   2.266 0.023901 *
## as.factor(rad)6   1.497927   1.560891   0.960 0.337698
## as.factor(rad)7   3.426066   1.707781   2.006 0.045388 *
## as.factor(rad)8   4.971560   1.617701   3.073 0.002235 **
## as.factor(rad)24  5.018556   1.471749   3.410 0.000703 ***
## ptratio          -1.092511   0.155188  -7.040 6.52e-12 ***
## lstat            -0.537326   0.051082 -10.519  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.134 on 491 degrees of freedom
## Multiple R-squared:  0.697,  Adjusted R-squared:  0.6884
## F-statistic: 80.69 on 14 and 491 DF,  p-value: < 2.2e-16
```

```
# And from a backwards selection of the Boston.P4update I will remove zn from our model
 since it is also not significant
Boston.P4Update2<- update(Boston.P4Update,~.-zn)
summary(Boston.P4Update2)
```

```
##
## Call:
## lm(formula = medv ~ crim + nox + rm + as.factor(rad) + ptratio +
##     lstat, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.190  -3.077  -0.792   1.895  28.975
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      20.37551    4.71969   4.317 1.91e-05 ***
## crim             -0.09693    0.03518  -2.755 0.006087 **
## nox              -3.22554    3.01812  -1.069 0.285718
## rm                4.27712    0.43773   9.771  < 2e-16 ***
## as.factor(rad)2   3.27544    1.56097   2.098 0.036385 *
## as.factor(rad)3   5.44426    1.42192   3.829 0.000145 ***
## as.factor(rad)4   3.25228    1.27535   2.550 0.011071 *
## as.factor(rad)5   3.13387    1.28380   2.441 0.014995 *
## as.factor(rad)6   1.63192    1.54036   1.059 0.289921
## as.factor(rad)7   3.51028    1.69958   2.065 0.039411 *
## as.factor(rad)8   5.18316    1.56943   3.303 0.001028 **
## as.factor(rad)24  5.04332    1.47000   3.431 0.000652 ***
## ptratio          -1.05905    0.14246  -7.434 4.71e-13 ***
## lstat            -0.53623    0.05101 -10.513  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.13 on 492 degrees of freedom
## Multiple R-squared:  0.6968, Adjusted R-squared:  0.6888
## F-statistic: 86.99 on 13 and 492 DF,  p-value: < 2.2e-16
```

```
#Now I will start to add polynomial terms. First I would like to look at correlations.

cor(crim,crim^2)
```

```
## [1] 0.8710611
```

```
cor(nox,nox^2)
```

```
## [1] 0.9935007
```

```
cor(lstat,lstat^2)
```

```
## [1] 0.9605726
```

```
cor(rm,rm^2)
```

```
## [1] 0.994528
```

```
cor(ptratio,ptratio^2)
```

```
## [1] 0.9979917
```

```
#the correlation between all of the above variables and their square could be a problem
#lets try the square of the transformed variables
tcrim<-(Boston$crim-mean(Boston$crim))/sd(Boston$crim)
tlstat<-(Boston$lstat-mean(Boston$lstat))/sd(Boston$lstat)


cor(crim,tcrim^2)
```

```
## [1] 0.8365473
```

```
cor(lstat,tlstat^2)
```

```
## [1] 0.5742952
```

```
Boston.Poly<- update(Boston.P4Update2,~.+I(tlstat^2)+I(tcrim^2)+crim*age+rm*tax+rm*ptrat
io+nox*crim+nox*age+indus*tax+crim*tax+ptratio*crim, data = Boston)
summary(Boston.Poly)
```

```
##
## Call:
## lm(formula = medv ~ crim + nox + rm + as.factor(rad) + ptratio +
##     lstat + I(tlstat^2) + I(tcrim^2) + age + tax + indus + crim:age +
##     rm:tax + rm:ptratio + crim:nox + nox:age + tax:indus + crim:tax +
##     crim:ptratio, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.7767 -2.4045 -0.3254  1.7169 26.8505
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -1.079e+02  1.602e+01  -6.738 4.62e-11 ***
## crim              4.437e+00  2.491e+00   1.781 0.075499 .
## nox               2.007e+01  1.066e+01   1.883 0.060293 .
## rm                2.187e+01  2.253e+00   9.703  < 2e-16 ***
## as.factor(rad)2   1.290e+00  1.268e+00   1.018 0.309395
## as.factor(rad)3   4.954e+00  1.139e+00   4.349 1.67e-05 ***
## as.factor(rad)4   2.632e+00  1.041e+00   2.528 0.011806 *
## as.factor(rad)5   2.407e+00  1.037e+00   2.321 0.020730 *
## as.factor(rad)6   2.947e+00  1.270e+00   2.320 0.020755 *
## as.factor(rad)7   3.780e+00  1.341e+00   2.819 0.005011 **
## as.factor(rad)8   3.483e+00  1.278e+00   2.725 0.006656 **
## as.factor(rad)24  9.190e+00  1.692e+00   5.431 8.93e-08 ***
## ptratio           4.248e+00  9.832e-01   4.320 1.90e-05 ***
## lstat            -7.789e-01  5.852e-02 -13.310  < 2e-16 ***
## I(tlstat^2)       1.446e+00  1.817e-01   7.957 1.29e-14 ***
## I(tcrim^2)        3.035e-01  8.347e-02   3.636 0.000307 ***
## age               1.458e-01  5.775e-02   2.525 0.011906 *
## tax               5.397e-02  1.685e-02   3.204 0.001448 **
## indus             2.565e-01  1.164e-01   2.204 0.028024 *
## crim:age          4.782e-03  3.175e-03   1.506 0.132693
## rm:tax           -9.858e-03  2.339e-03  -4.215 2.99e-05 ***
## rm:ptratio       -7.737e-01  1.473e-01  -5.254 2.25e-07 ***
## crim:nox         -2.053e+00  6.431e-01  -3.192 0.001505 **
## nox:age          -2.575e-01  1.211e-01  -2.126 0.034015 *
## tax:indus        -2.877e-04  3.036e-04  -0.947 0.343910
## crim:tax          2.733e-04  2.850e-03   0.096 0.923647
## crim:ptratio     -2.042e-01  1.798e-01  -1.136 0.256573
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4 on 479 degrees of freedom
## Multiple R-squared:  0.8206, Adjusted R-squared:  0.8109
## F-statistic: 84.28 on 26 and 479 DF,  p-value: < 2.2e-16
```

```
#Now I will do a step wise to find the best model.

step3 <- stepAIC(Boston.Poly, direction="both")
```

```
## Start:  AIC=1429.1
## medv ~ crim + nox + rm + as.factor(rad) + ptratio + lstat + I(tlstat^2) +
##     I(tcrim^2) + age + tax + indus + crim:age + rm:tax + rm:ptratio +
##     crim:nox + nox:age + tax:indus + crim:tax + crim:ptratio
##
##                  Df Sum of Sq     RSS    AIC
## - crim:tax        1      0.15  7662.9 1427.1
## - tax:indus       1     14.36  7677.1 1428.0
## - crim:ptratio    1     20.64  7683.4 1428.5
## <none>                         7662.8 1429.1
## - crim:age        1     36.29  7699.1 1429.5
## - nox:age         1     72.31  7735.1 1431.8
## - crim:nox        1    163.01  7825.8 1437.8
## - I(tcrim^2)      1    211.48  7874.3 1440.9
## - rm:tax          1    284.22  7947.0 1445.5
## - rm:ptratio      1    441.53  8104.3 1455.5
## - as.factor(rad)  8    813.00  8475.8 1464.1
## - I(tlstat^2)     1   1012.76  8675.5 1489.9
## - lstat           1   2834.24 10497.0 1586.3
##
## Step:  AIC=1427.11
## medv ~ crim + nox + rm + as.factor(rad) + ptratio + lstat + I(tlstat^2) +
##     I(tcrim^2) + age + tax + indus + crim:age + rm:tax + rm:ptratio +
##     crim:nox + nox:age + tax:indus + crim:ptratio
##
##                  Df Sum of Sq     RSS    AIC
## - tax:indus       1     14.22  7677.2 1426.0
## <none>                         7662.9 1427.1
## - crim:age        1     37.12  7700.1 1427.6
## - crim:ptratio    1     42.87  7705.8 1427.9
## + crim:tax        1      0.15  7662.8 1429.1
## - nox:age         1     72.16  7735.1 1429.9
## - crim:nox        1    166.90  7829.8 1436.0
## - I(tcrim^2)      1    211.77  7874.7 1438.9
## - rm:tax          1    297.82  7960.8 1444.4
## - rm:ptratio      1    466.64  8129.6 1455.0
## - as.factor(rad)  8    832.13  8495.1 1463.3
## - I(tlstat^2)     1   1015.19  8678.1 1488.1
## - lstat           1   2844.73 10507.7 1584.9
##
## Step:  AIC=1426.05
## medv ~ crim + nox + rm + as.factor(rad) + ptratio + lstat + I(tlstat^2) +
##     I(tcrim^2) + age + tax + indus + crim:age + rm:tax + rm:ptratio +
##     crim:nox + nox:age + crim:ptratio
##
##                  Df Sum of Sq     RSS    AIC
## <none>                         7677.2 1426.0
## - crim:age        1     37.26  7714.4 1426.5
## - crim:ptratio    1     45.42  7722.6 1427.0
## + tax:indus       1     14.22  7662.9 1427.1
## + crim:tax        1      0.01  7677.1 1428.0
## - nox:age         1     68.43  7745.6 1428.5
## - indus           1    138.61  7815.8 1433.1
```

```
## - crim:nox        1    167.04  7844.2 1434.9
## - I(tcrim^2)      1    205.74  7882.9 1437.4
## - rm:tax          1    285.99  7963.1 1442.6
## - rm:ptratio      1    489.34  8166.5 1455.3
## - as.factor(rad)  8    821.20  8498.4 1461.5
## - I(tlstat^2)     1   1017.94  8695.1 1487.0
## - lstat           1   2862.08 10539.2 1584.4
```

```
step3$anova # display results
```

```
## Stepwise Model Path 
## Analysis of Deviance Table
## 
## Initial Model:
## medv ~ crim + nox + rm + as.factor(rad) + ptratio + lstat + I(tlstat^2) + 
##     I(tcrim^2) + age + tax + indus + crim:age + rm:tax + rm:ptratio + 
##     crim:nox + nox:age + tax:indus + crim:tax + crim:ptratio
## 
## Final Model:
## medv ~ crim + nox + rm + as.factor(rad) + ptratio + lstat + I(tlstat^2) + 
##     I(tcrim^2) + age + tax + indus + crim:age + rm:tax + rm:ptratio + 
##     crim:nox + nox:age + crim:ptratio
## 
## 
##          Step Df   Deviance Resid. Df Resid. Dev      AIC
## 1                                 479   7662.785 1429.103
## 2  - crim:tax  1  0.1471001       480   7662.932 1427.112
## 3 - tax:indus  1 14.2218170       481   7677.154 1426.051
```

```
BestPY<- lm(medv ~ crim + nox + rm + as.factor(rad) + ptratio + lstat + I(tlstat^2) + 
              I(tcrim^2) + age + tax + indus + crim*age + rm*tax + rm*ptratio + 
              crim*nox + nox*age + crim*ptratio, data = Boston)
```

e. **Compare** the performance of the models in part (a) to (d). Look at various diagnostics we have seen. Also check for normality and constant variance violations. Make a comparison and support your comments with plots and statistics.

Answer goes here:

```
print(paste0("Boston Base Model adjusted R squared   ", summary(Boston.BM)$adj.r.squared
,"  Boston Base Model residual standard error   ",  summary(Boston.BM)$sigma) )
```

```
## [1] "Boston Base Model adjusted R squared   0.733408239842023  Boston Base Model resi
dual standard error   4.7486970348662"
```

```
print(paste0("Boston Interaction Model adjusted R squared   ", summary(Boston.ITF)$adj.
r.squared,"  Boston Interaction Model residual standard error   ",  summary(Boston.ITF)
$sigma) )
```

```
## [1] "Boston Interaction Model adjusted R squared   0.767925400503753  Boston Interact
ion Model residual standard error   4.43062410962611"
```

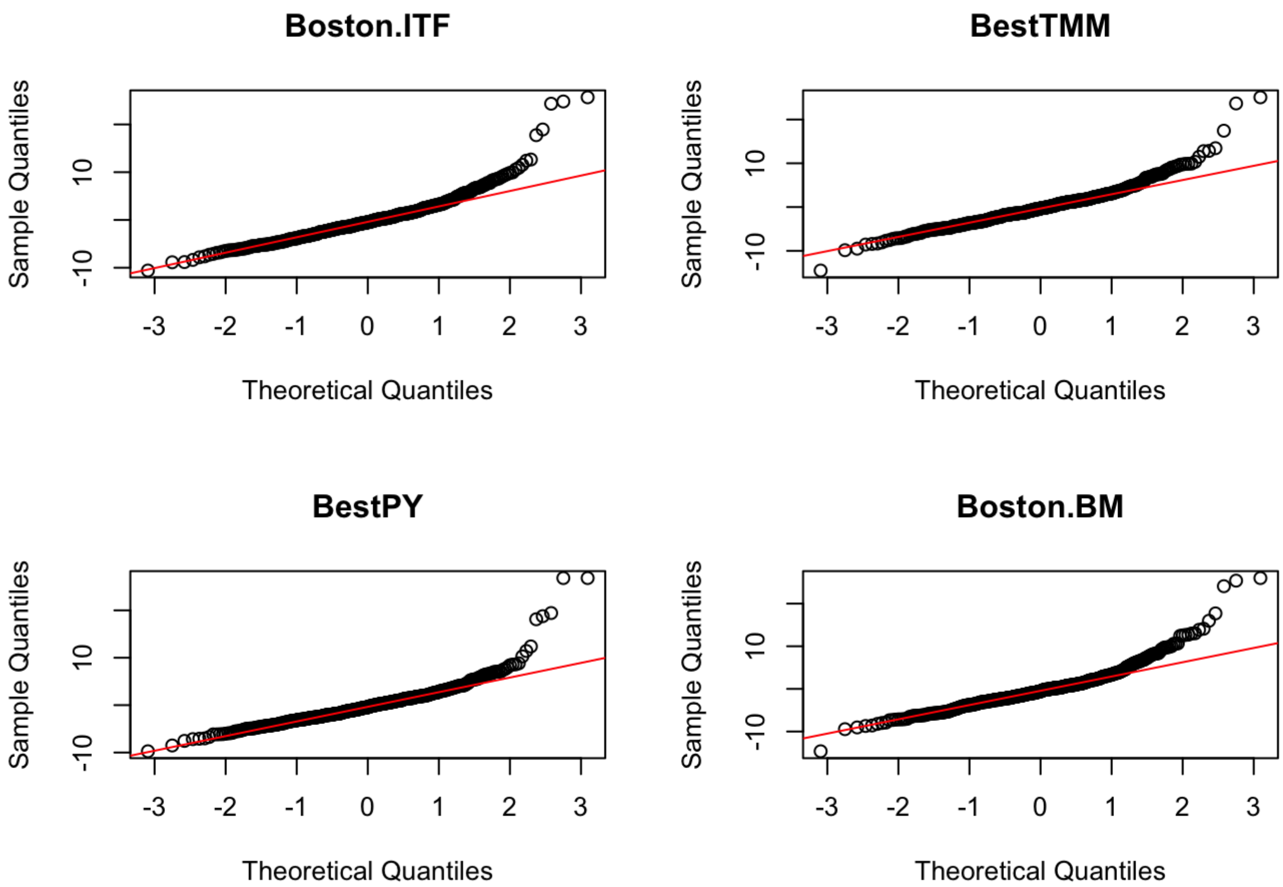
```
print(paste0("Boston Polynomial Model adjusted R squared   ", summary(BestPY)$adj.r.squa
red, "  Boston Polynomial Model residual standard error   ",  summary(BestPY)$sigma))
```

```
## [1] "Boston Polynomial Model adjusted R squared   0.811308222633533  Boston Polynomia
l Model residual standard error   3.99509940008713"
```

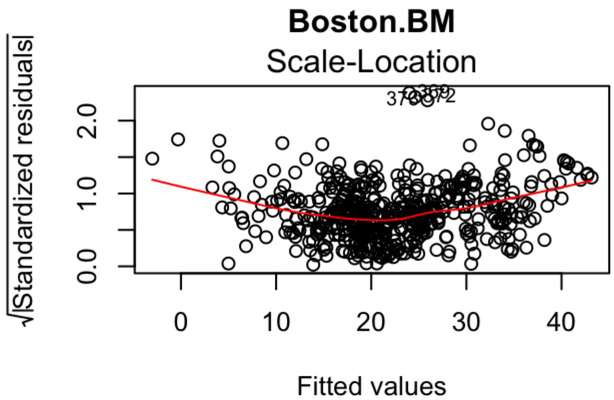
```
print(paste0("Boston Transfromation Model adjusted R squared   ", summary(BestTMM)$adj.
r.squared,  "  Boston Transformation Model residual standard error   ",  summary(BestTM
M)$sigma) )
```

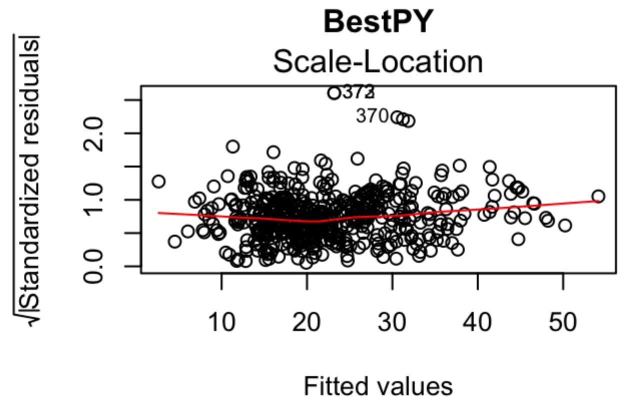
```
## [1] "Boston Transfromation Model adjusted R squared   0.785837164391323  Boston Trans
formation Model residual standard error   4.25621105911814"
```

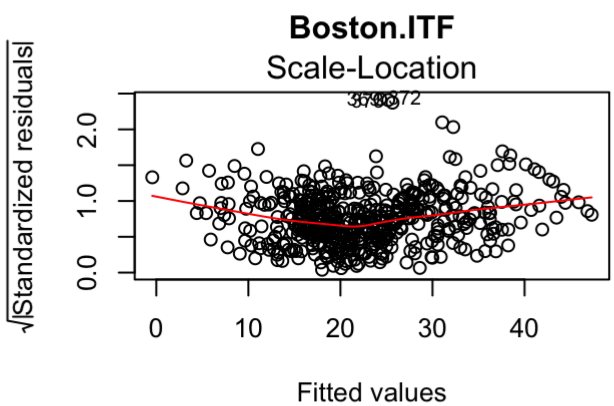
```
#testing for normaility
layout(matrix(c(1,2,3,4),2,2))
qqnorm(residuals(Boston.ITF), main='Boston.ITF')
qqline(residuals(Boston.ITF), col='red')
qqnorm(residuals(BestPY), main='BestPY')
qqline(residuals(BestPY), col='red')
qqnorm(residuals(BestTMM), main='BestTMM')
qqline(residuals(BestTMM), col='red')
qqnorm(residuals(Boston.BM), main='Boston.BM')
qqline(residuals(Boston.BM), col='red')
```

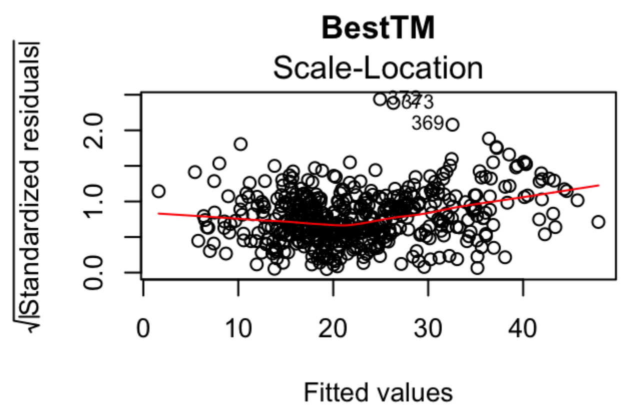



```
#Looking at Standardized residuals to determine if there is a constant variance
layout(matrix(c(1,2,3,4),2,2))
plot(Boston.BM,3, main='Boston.BM')
plot(Boston.ITF, 3, main='Boston.ITF')
plot(BestPY, 3, main='BestPY')
plot(BestTMM,3, main='BestTM')
```









From the above comparisons we can see that the best fitted model is The Boston Polynomial Model (BestPY). It is the best fitted model since it has the highest adjusted R^2 and standard error. It is also the most normal model and seems to have a constant variance.

f. **Make your own:** Now considering all of the above, propose a new model different than the ones in part a-d (try mixture of the suggestions above). Use best subsets to fit your model. Comment on overall adequacy of your model comparing with the ones above.

Answer goes here (model, summary, explanation and comparison):

```
library(leaps)
model.subset <- regsubsets(log(medv)~crim+zn+indus+as.factor(chas)+nox+rm+age+dis+as.fac
tor(rad)+tax+ptratio+lstat+ I(tlstat^2) + crim*age + rm*tax + rm*ptratio + crim*nox + no
x*age + crim*ptratio, data = Boston, nbest = 1, nvmax = 26)
summary(model.subset)
```

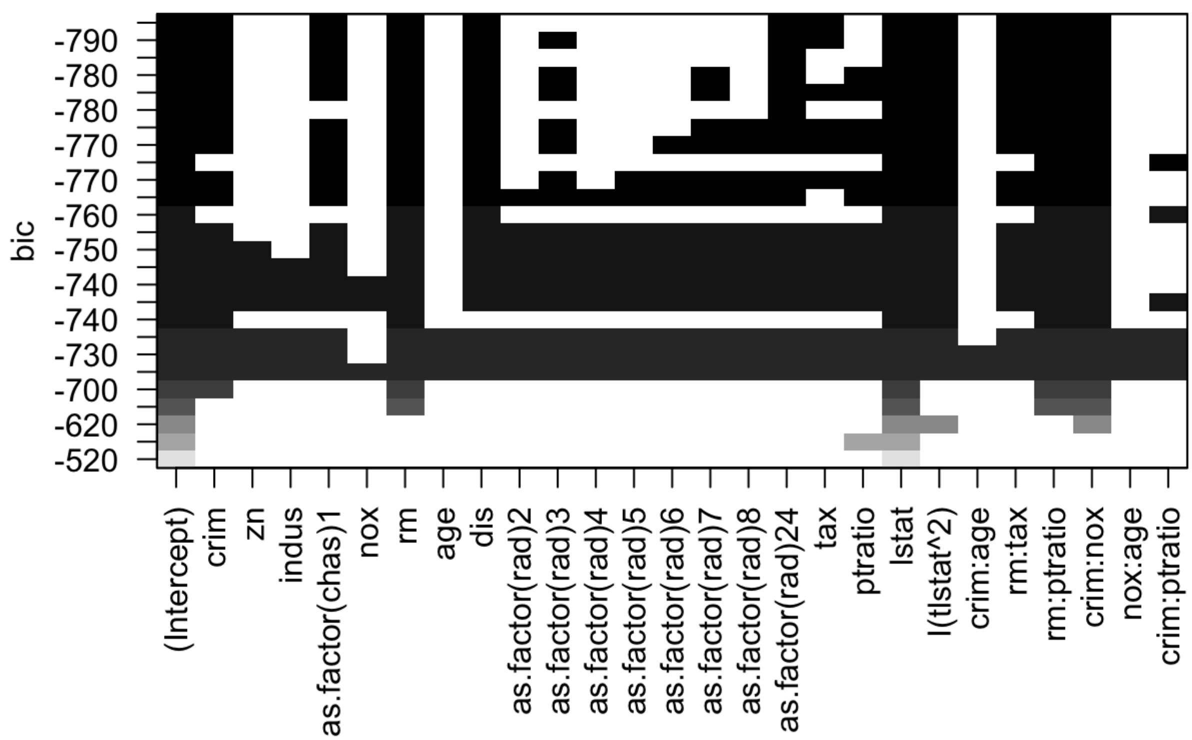
```
## Subset selection object
## Call: regsubsets.formula(log(medv) ~ crim + zn + indus + as.factor(chas) +
##     nox + rm + age + dis + as.factor(rad) + tax + ptratio + lstat +
##     I(tlstat^2) + crim * age + rm * tax + rm * ptratio + crim *
##     nox + nox * age + crim * ptratio, data = Boston, nbest = 1,
##     nvmax = 26)
## 26 Variables  (and intercept)
##                  Forced in Forced out
## crim                 FALSE      FALSE
## zn                   FALSE      FALSE
## indus                FALSE      FALSE
## as.factor(chas)1     FALSE      FALSE
## nox                  FALSE      FALSE
## rm                   FALSE      FALSE
## age                  FALSE      FALSE
## dis                  FALSE      FALSE
## as.factor(rad)2      FALSE      FALSE
## as.factor(rad)3      FALSE      FALSE
## as.factor(rad)4      FALSE      FALSE
## as.factor(rad)5      FALSE      FALSE
## as.factor(rad)6      FALSE      FALSE
## as.factor(rad)7      FALSE      FALSE
## as.factor(rad)8      FALSE      FALSE
## as.factor(rad)24     FALSE      FALSE
## tax                  FALSE      FALSE
## ptratio              FALSE      FALSE
## lstat                FALSE      FALSE
## I(tlstat^2)          FALSE      FALSE
## crim:age             FALSE      FALSE
## rm:tax               FALSE      FALSE
## rm:ptratio           FALSE      FALSE
## crim:nox             FALSE      FALSE
## nox:age              FALSE      FALSE
## crim:ptratio         FALSE      FALSE
## 1 subsets of each size up to 26
## Selection Algorithm: exhaustive
##           crim zn  indus as.factor(chas)1 nox rm  age dis as.factor(rad)2
## 1  ( 1 )  " "  " " " "   " "              " " " " " " " " " "
## 2  ( 1 )  " "  " " " "   " "              " " " " " " " " " "
## 3  ( 1 )  " "  " " " "   " "              " " " " " " " " " "
## 4  ( 1 )  " "  " " " "   " "              " " "*" " " " " " "
## 5  ( 1 )  "*"  " " " "   " "              " " "*" " " " " " "
## 6  ( 1 )  "*"  " " " "   " "              " " "*" " " " " " "
## 7  ( 1 )  " "  " " " "   " "              " " "*" " " "*" " "
## 8  ( 1 )  " "  " " " "   "*"              " " "*" " " "*" " "
## 9  ( 1 )  "*"  " " " "   " "              " " "*" " " "*" " "
## 10  ( 1 ) "*"  " " " "   "*"              " " "*" " " "*" " "
## 11  ( 1 ) "*"  " " " "   "*"              " " "*" " " "*" " "
## 12  ( 1 ) "*"  " " " "   "*"              " " "*" " " "*" " "
## 13  ( 1 ) "*"  " " " "   "*"              " " "*" " " "*" " "
## 14  ( 1 ) "*"  " " " "   "*"              " " "*" " " "*" " "
## 15  ( 1 ) "*"  " " " "   "*"              " " "*" " " "*" " "
## 16  ( 1 ) "*"  " " " "   "*"              " " "*" " " "*" " "
```

```
## 17  ( 1 ) "*"  " " " "   "*"              " " "*" " " "*" " "
## 18  ( 1 ) "*"  " " " "   "*"              " " "*" " " "*" "*"
## 19  ( 1 ) "*"  " " " "   "*"              " " "*" " " "*" "*"
## 20  ( 1 ) "*"  "*" " "   "*"              " " "*" " " "*" "*"
## 21  ( 1 ) "*"  "*" "*"   "*"              " " "*" " " "*" "*"
## 22  ( 1 ) "*"  "*" "*"   "*"              "*" "*" " " "*" "*"
## 23  ( 1 ) "*"  "*" "*"   "*"              "*" "*" " " "*" "*"
## 24  ( 1 ) "*"  "*" "*"   "*"              " " "*" "*" "*" "*"
## 25  ( 1 ) "*"  "*" "*"   "*"              " " "*" "*" "*" "*"
## 26  ( 1 ) "*"  "*" "*"   "*"              "*" "*" "*" "*" "*"
##           as.factor(rad)3 as.factor(rad)4 as.factor(rad)5 as.factor(rad)6
## 1  ( 1 )  " "             " "             " "             " "
## 2  ( 1 )  " "             " "             " "             " "
## 3  ( 1 )  " "             " "             " "             " "
## 4  ( 1 )  " "             " "             " "             " "
## 5  ( 1 )  " "             " "             " "             " "
## 6  ( 1 )  " "             " "             " "             " "
## 7  ( 1 )  " "             " "             " "             " "
## 8  ( 1 )  " "             " "             " "             " "
## 9  ( 1 )  " "             " "             " "             " "
## 10  ( 1 ) " "             " "             " "             " "
## 11  ( 1 ) " "             " "             " "             " "
## 12  ( 1 ) "*"             " "             " "             " "
## 13  ( 1 ) "*"             " "             " "             " "
## 14  ( 1 ) "*"             " "             " "             " "
## 15  ( 1 ) "*"             " "             " "             " "
## 16  ( 1 ) "*"             " "             " "             "*"
## 17  ( 1 ) "*"             " "             "*"             "*"
## 18  ( 1 ) "*"             "*"             "*"             "*"
## 19  ( 1 ) "*"             "*"             "*"             "*"
## 20  ( 1 ) "*"             "*"             "*"             "*"
## 21  ( 1 ) "*"             "*"             "*"             "*"
## 22  ( 1 ) "*"             "*"             "*"             "*"
## 23  ( 1 ) "*"             "*"             "*"             "*"
## 24  ( 1 ) "*"             "*"             "*"             "*"
## 25  ( 1 ) "*"             "*"             "*"             "*"
## 26  ( 1 ) "*"             "*"             "*"             "*"
##           as.factor(rad)7 as.factor(rad)8 as.factor(rad)24 tax ptratio
## 1  ( 1 )  " "             " "             " "              " " " "
## 2  ( 1 )  " "             " "             " "              " " "*"
## 3  ( 1 )  " "             " "             " "              " " " "
## 4  ( 1 )  " "             " "             " "              " " " "
## 5  ( 1 )  " "             " "             " "              " " " "
## 6  ( 1 )  " "             " "             " "              " " " "
## 7  ( 1 )  " "             " "             " "              " " " "
## 8  ( 1 )  " "             " "             " "              " " " "
## 9  ( 1 )  " "             " "             "*"              " " " "
## 10  ( 1 ) " "             " "             "*"              " " " "
## 11  ( 1 ) " "             " "             "*"              "*" " "
## 12  ( 1 ) " "             " "             "*"              "*" " "
## 13  ( 1 ) "*"             " "             "*"              " " "*"
## 14  ( 1 ) "*"             " "             "*"              "*" "*"
## 15  ( 1 ) "*"             "*"             "*"              "*" "*"
## 16  ( 1 ) "*"             "*"             "*"              "*" "*"
```

```
## 17  ( 1 ) "*"              "*"              "*"               "*" "*"
## 18  ( 1 ) "*"              "*"              "*"               " " "*"
## 19  ( 1 ) "*"              "*"              "*"               "*" "*"
## 20  ( 1 ) "*"              "*"              "*"               "*" "*"
## 21  ( 1 ) "*"              "*"              "*"               "*" "*"
## 22  ( 1 ) "*"              "*"              "*"               "*" "*"
## 23  ( 1 ) "*"              "*"              "*"               "*" "*"
## 24  ( 1 ) "*"              "*"              "*"               "*" "*"
## 25  ( 1 ) "*"              "*"              "*"               "*" "*"
## 26  ( 1 ) "*"              "*"              "*"               "*" "*"
##           lstat I(tlstat^2) crim:age rm:tax rm:ptratio crim:nox nox:age
## 1  ( 1 )  "*"   " "         " "      " "    " "        " "      " "    
## 2  ( 1 )  "*"   " "         " "      " "    " "        " "      " "    
## 3  ( 1 )  "*"   "*"         " "      " "    " "        "*"      " "    
## 4  ( 1 )  "*"   " "         " "      " "    "*"        "*"      " "    
## 5  ( 1 )  "*"   " "         " "      " "    "*"        "*"      " "    
## 6  ( 1 )  "*"   "*"         " "      " "    "*"        "*"      " "    
## 7  ( 1 )  "*"   "*"         " "      " "    "*"        "*"      " "    
## 8  ( 1 )  "*"   "*"         " "      " "    "*"        "*"      " "    
## 9  ( 1 )  "*"   "*"         " "      "*"    "*"        "*"      " "    
## 10  ( 1 ) "*"   "*"         " "      "*"    "*"        "*"      " "    
## 11  ( 1 ) "*"   "*"         " "      "*"    "*"        "*"      " "    
## 12  ( 1 ) "*"   "*"         " "      "*"    "*"        "*"      " "    
## 13  ( 1 ) "*"   "*"         " "      "*"    "*"        "*"      " "    
## 14  ( 1 ) "*"   "*"         " "      "*"    "*"        "*"      " "    
## 15  ( 1 ) "*"   "*"         " "      "*"    "*"        "*"      " "    
## 16  ( 1 ) "*"   "*"         " "      "*"    "*"        "*"      " "    
## 17  ( 1 ) "*"   "*"         " "      "*"    "*"        "*"      " "    
## 18  ( 1 ) "*"   "*"         " "      "*"    "*"        "*"      " "    
## 19  ( 1 ) "*"   "*"         " "      "*"    "*"        "*"      " "    
## 20  ( 1 ) "*"   "*"         " "      "*"    "*"        "*"      " "    
## 21  ( 1 ) "*"   "*"         " "      "*"    "*"        "*"      " "    
## 22  ( 1 ) "*"   "*"         " "      "*"    "*"        "*"      " "    
## 23  ( 1 ) "*"   "*"         " "      "*"    "*"        "*"      " "    
## 24  ( 1 ) "*"   "*"         " "      "*"    "*"        "*"      "*"    
## 25  ( 1 ) "*"   "*"         "*"      "*"    "*"        "*"      "*"    
## 26  ( 1 ) "*"   "*"         "*"      "*"    "*"        "*"      "*"    
##           crim:ptratio
## 1  ( 1 )  " "         
## 2  ( 1 )  " "         
## 3  ( 1 )  " "         
## 4  ( 1 )  " "         
## 5  ( 1 )  " "         
## 6  ( 1 )  " "         
## 7  ( 1 )  "*"         
## 8  ( 1 )  "*"         
## 9  ( 1 )  " "         
## 10  ( 1 ) " "         
## 11  ( 1 ) " "         
## 12  ( 1 ) " "         
## 13  ( 1 ) " "         
## 14  ( 1 ) " "         
## 15  ( 1 ) " "         
## 16  ( 1 ) " "         
```

```
## 17  ( 1 ) " "
## 18  ( 1 ) " "
## 19  ( 1 ) " "
## 20  ( 1 ) " "
## 21  ( 1 ) " "
## 22  ( 1 ) " "
## 23  ( 1 ) "*"
## 24  ( 1 ) "*"
## 25  ( 1 ) "*"
## 26  ( 1 ) "*"
```

```
plot(model.subset, scale = "bic")
```



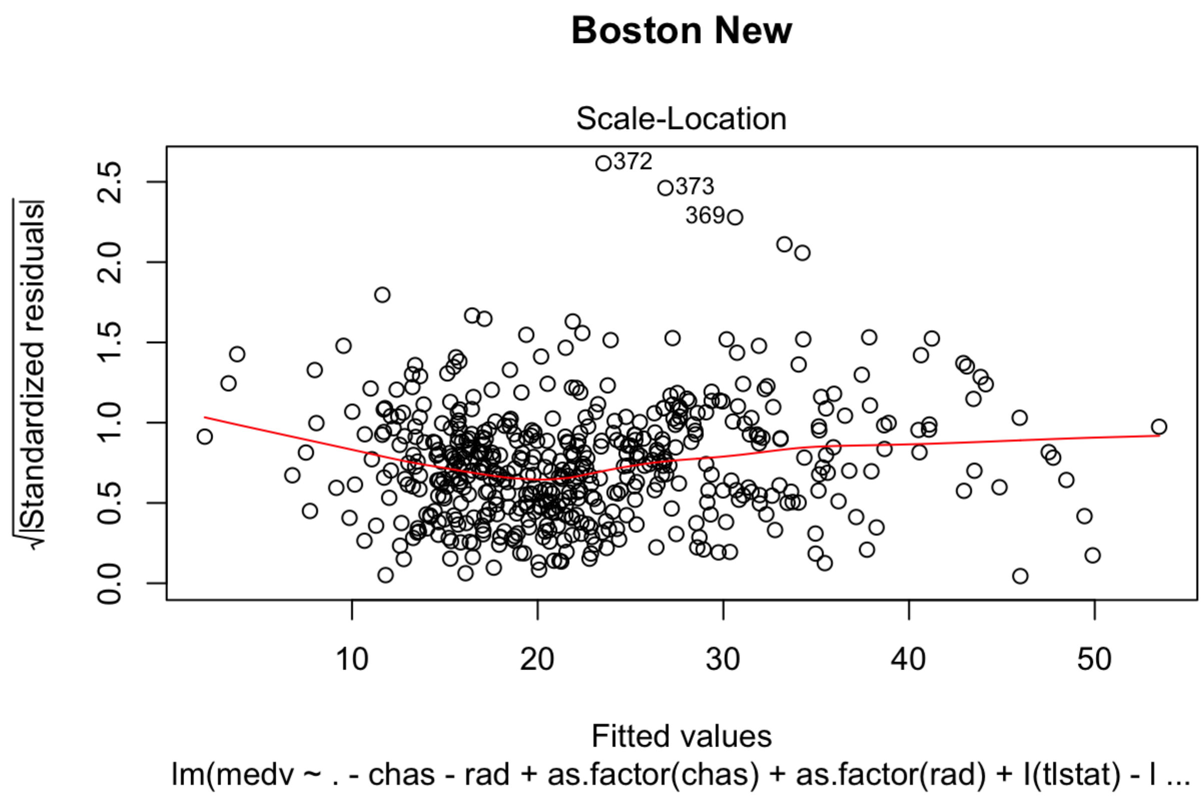

```
neww<- lm(medv~.-chas-rad+as.factor(chas)+as.factor(rad)+I(tlstat)-lstat+I(tlstat^2)-ind
us + rm*tax + rm*ptratio +
          crim*nox + nox*age + crim*ptratio-nox-zn, data = Boston)
summary(neww)
```

```
##
## Call:
## lm(formula = medv ~ . - chas - rad + as.factor(chas) + as.factor(rad) +
##     I(tlstat) - lstat + I(tlstat^2) - indus + rm * tax + rm *
##     ptratio + crim * nox + nox * age + crim * ptratio - nox -
##     zn, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.988 -2.357 -0.295  1.710 26.452
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -97.144965  14.593662  -6.657 7.63e-11 ***
## crim               4.877233   2.164966   2.253 0.024720 *
## rm                20.736829   2.146825   9.659  < 2e-16 ***
## age                0.044336   0.027795   1.595 0.111338
## dis               -0.713249   0.145456  -4.904 1.29e-06 ***
## tax                0.039872   0.013576   2.937 0.003473 **
## ptratio            4.358495   0.910488   4.787 2.25e-06 ***
## as.factor(chas)1   3.350686   0.720684   4.649 4.30e-06 ***
## as.factor(rad)2    1.747042   1.220218   1.432 0.152863
## as.factor(rad)3    4.350515   1.106542   3.932 9.67e-05 ***
## as.factor(rad)4    2.251578   0.989912   2.275 0.023372 *
## as.factor(rad)5    2.137704   1.003695   2.130 0.033690 *
## as.factor(rad)6    2.502674   1.221602   2.049 0.041033 *
## as.factor(rad)7    4.025847   1.307510   3.079 0.002195 **
## as.factor(rad)8    2.857172   1.226946   2.329 0.020287 *
## as.factor(rad)24   5.776371   1.495176   3.863 0.000127 ***
## I(tlstat)         -5.487842   0.403798 -13.591  < 2e-16 ***
## I(tlstat^2)        1.268794   0.172725   7.346 8.78e-13 ***
## rm:tax            -0.007640   0.002136  -3.576 0.000384 ***
## rm:ptratio        -0.776008   0.138199  -5.615 3.32e-08 ***
## crim:nox          -1.788827   0.614044  -2.913 0.003743 **
## nox:age           -0.077856   0.046162  -1.687 0.092331 .
## crim:ptratio      -0.190095   0.110191  -1.725 0.085141 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.91 on 483 degrees of freedom
## Multiple R-squared:  0.8271, Adjusted R-squared:  0.8193
## F-statistic: 105.1 on 22 and 483 DF,  p-value: < 2.2e-16
```

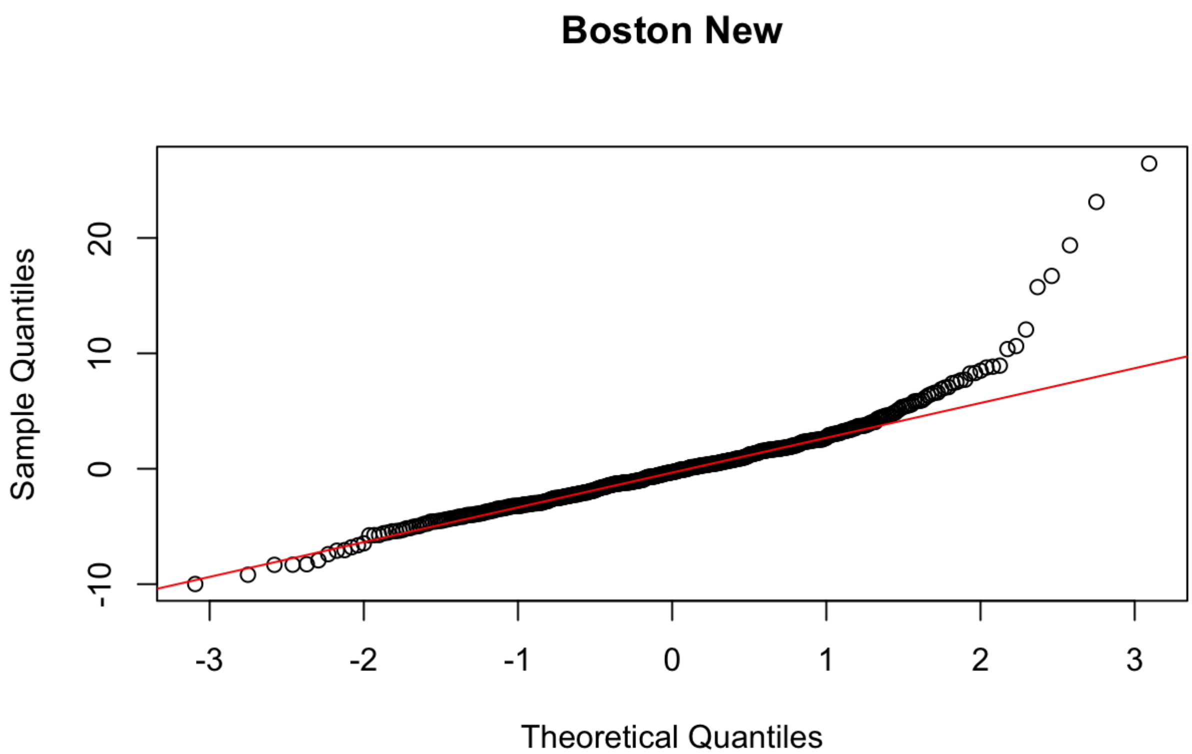
```
print(paste0("Boston New Model adjusted R squared   ", summary(neww)$adj.r.squared  ,"
 Boston New Model residual standard error   ",  summary(neww)$sigma) )
```

```
## [1] "Boston New Model adjusted R squared   0.819268765413949  Boston New Model residu
al standard error   3.90991853940085"
```

```
plot(neww,3, main='Boston New')
```



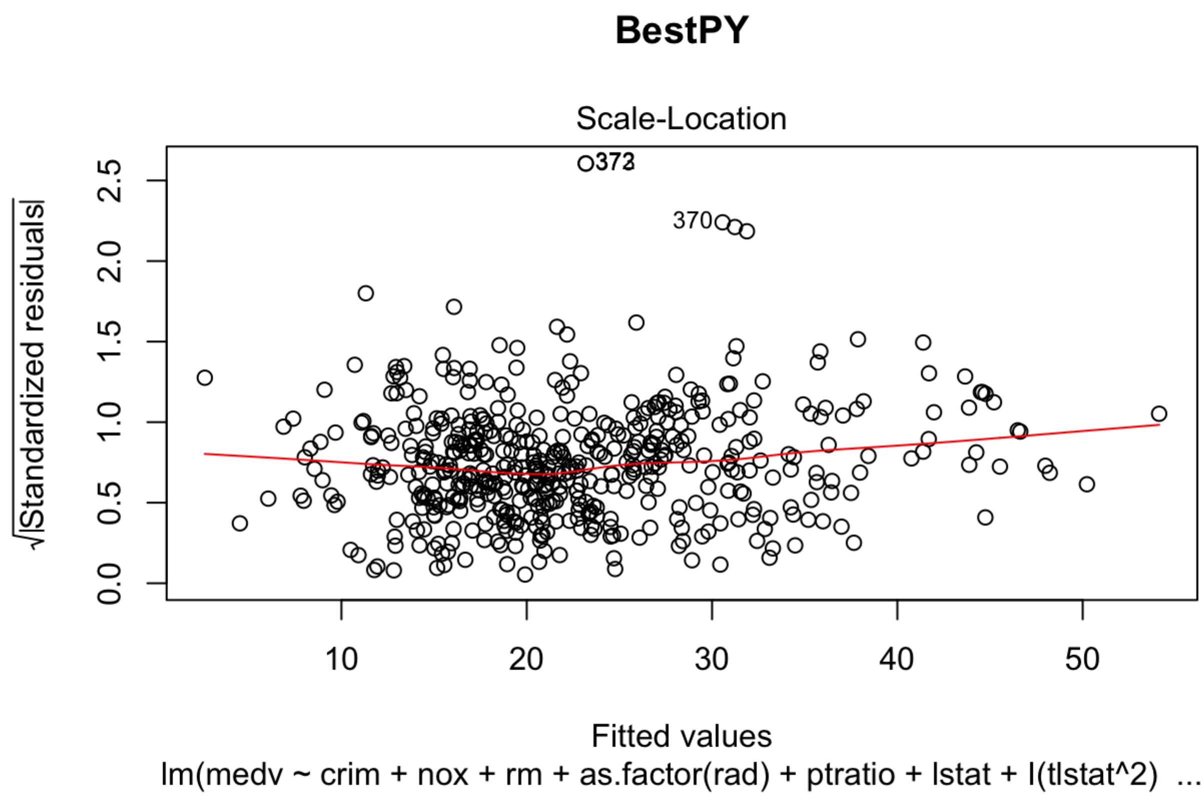

```
qqnorm(residuals(neww), main='Boston New')
qqline(residuals(neww), col='red')
```



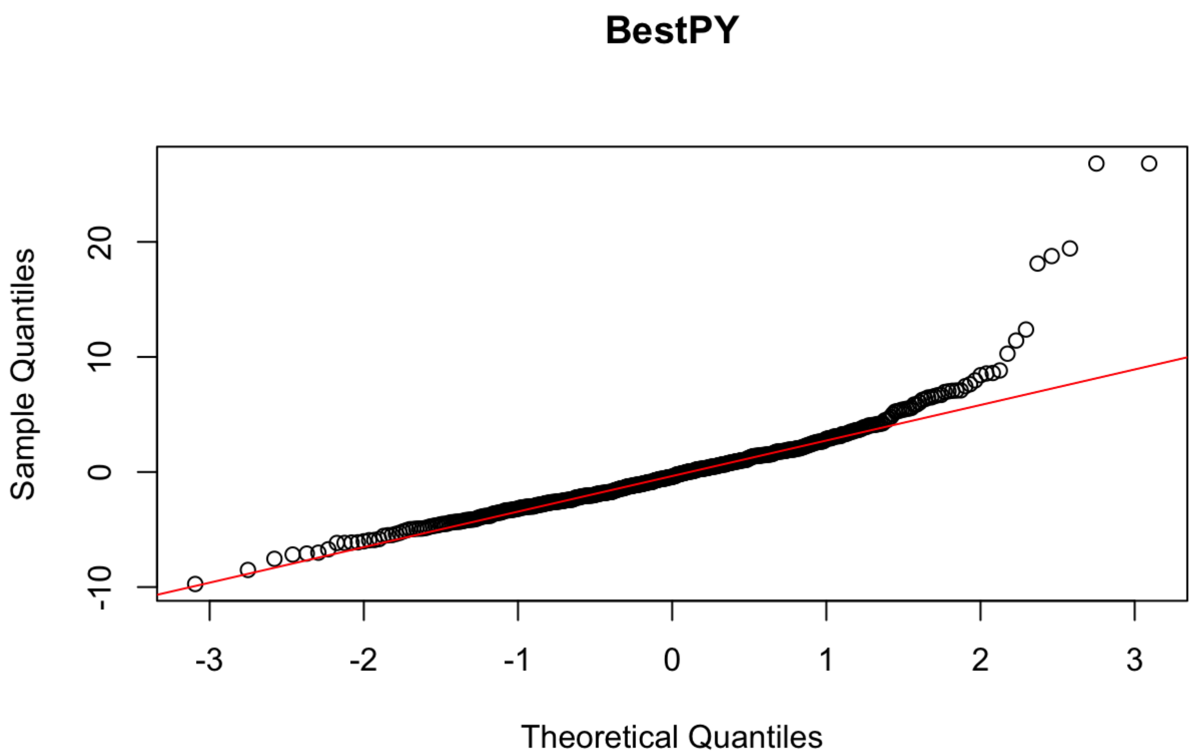

```
print(paste0("Boston Polynomial Model adjusted R squared   ", summary(BestPY)$adj.r.squa
red, "  Boston Polynomial Model residual standard error   ",  summary(BestPY)$sigma))
```

```
## [1] "Boston Polynomial Model adjusted R squared   0.811308222633533  Boston Polynomia
l Model residual standard error   3.99509940008713"
```

```
plot(BestPY, 3, main='BestPY')
```




```
qqnorm(residuals(BestPY), main='BestPY')
qqline(residuals(BestPY), col='red')
```



So,

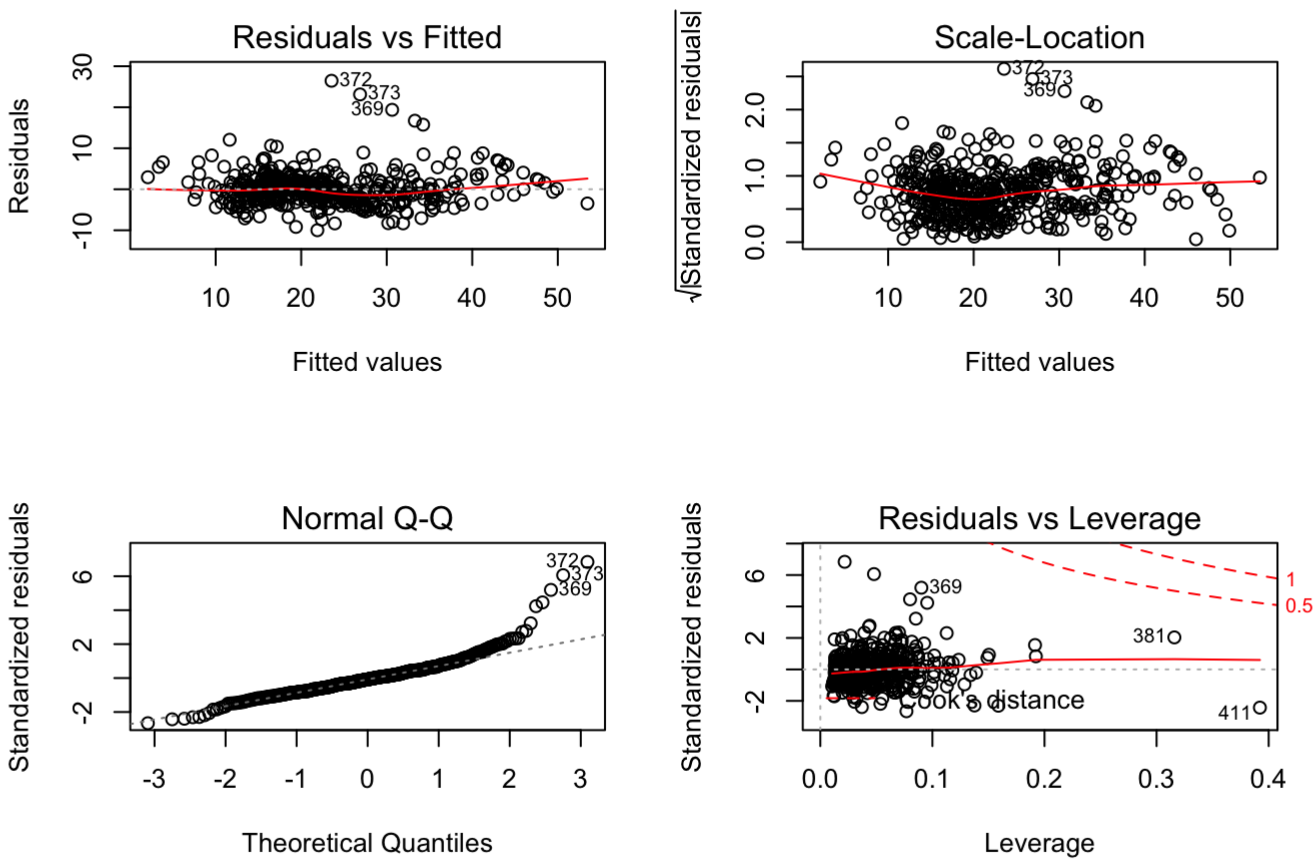
folllowing the KISS principle we will just go with the Boston Polynomial Model as it is the simpler model.

g. **Assesing the model:** Using your model in (f)

- detect 3 points from the data which you think are most probably outliers but not influential points.
- Detect pure leverage points and influential points (if no such points then say not detected, if there are more than 3 then write the most significant 3).
- Calculate the R-Student residuals at the points you find in this part.

Answer goes here:

```
layout(matrix(c(1,2,3,4),2,2))
plot(neww)
```




```
outlierTest(neww)
```

```
##     rstudent unadjusted p-value Bonferroni p
## 372 7.190132         2.4843e-12   1.2571e-09
## 373 6.298073         6.7948e-10   3.4382e-07
## 369 5.339081         1.4417e-07   7.2948e-05
## 370 4.545945         6.9246e-06   3.5038e-03
## 371 4.310024         1.9800e-05   1.0019e-02
```

I believe that the outliers are points 369,372, and 373. The most influential point are observations 411 and 381 with a large Cooks distance and the large leverage.

h. **Colinearity;** Check for multicollinearity in model part (a), part (d), and your model in part (f). Compare the differences in multicollinearity and discuss its possible causes.

Answer goes here:

```
# Evaluate Collinearity
vif(Boston.BM) # variance inflation factors
```

```
##                        GVIF Df GVIF^(1/(2*Df))
## crim               1.788890  1        1.337494
## zn                 2.495788  1        1.579806
## indus              4.366272  1        2.089563
## as.factor(chas)    1.099046  1        1.048354
## nox                4.653666  1        2.157236
## rm                 1.985990  1        1.409252
## age                3.164006  1        1.778765
## dis                4.141903  1        2.035167
## as.factor(rad)    18.629578  8        1.200571
## tax                9.869994  1        3.141655
## ptratio            2.241516  1        1.497169
## lstat              2.922144  1        1.709428
```

```
sqrt(vif(Boston.BM)) > 2 # problem?
```

```
##                  GVIF    Df GVIF^(1/(2*Df))
## crim            FALSE FALSE           FALSE
## zn              FALSE FALSE           FALSE
## indus            TRUE FALSE           FALSE
## as.factor(chas) FALSE FALSE           FALSE
## nox              TRUE FALSE           FALSE
## rm              FALSE FALSE           FALSE
## age             FALSE FALSE           FALSE
## dis              TRUE FALSE           FALSE
## as.factor(rad)   TRUE  TRUE           FALSE
## tax              TRUE FALSE           FALSE
## ptratio         FALSE FALSE           FALSE
## lstat           FALSE FALSE           FALSE
```

```
vif(BestPY) # variance inflation factors
```

```
##                          GVIF Df GVIF^(1/(2*Df))
## crim            12239.606245  1      110.632754
## nox                47.130197  1        6.865144
## rm                 75.579221  1        8.693631
## as.factor(rad)     30.654748  8        1.238529
## ptratio           129.795219  1       11.392770
## lstat               5.490035  1        2.343082
## I(tlstat^2)         2.564882  1        1.601525
## I(tcrim^2)          8.469944  1        2.910317
## age                82.609374  1        9.088970
## tax               172.425690  1       13.131096
## indus               4.358181  1        2.087626
## crim:age          213.762479  1       14.620618
## rm:tax            152.238138  1       12.338482
## rm:ptratio        145.439929  1       12.059848
## crim:nox          426.080019  1       20.641706
## nox:age           218.259411  1       14.773605
## crim:ptratio    12986.607978  1      113.958799
```

```
sqrt(vif(BestPY)) > 2 # problem?
```

```
##                  GVIF    Df GVIF^(1/(2*Df))
## crim             TRUE FALSE            TRUE
## nox              TRUE FALSE            TRUE
## rm               TRUE FALSE            TRUE
## as.factor(rad)   TRUE  TRUE           FALSE
## ptratio          TRUE FALSE            TRUE
## lstat            TRUE FALSE           FALSE
## I(tlstat^2)     FALSE FALSE           FALSE
## I(tcrim^2)       TRUE FALSE           FALSE
## age              TRUE FALSE            TRUE
## tax              TRUE FALSE            TRUE
## indus            TRUE FALSE           FALSE
## crim:age         TRUE FALSE            TRUE
## rm:tax           TRUE FALSE            TRUE
## rm:ptratio       TRUE FALSE            TRUE
## crim:nox         TRUE FALSE            TRUE
## nox:age          TRUE FALSE            TRUE
## crim:ptratio     TRUE FALSE            TRUE
```

```
vif(neww) # variance inflation factors
```

```
##                          GVIF Df GVIF^(1/(2*Df))
## crim             11455.416198  1      107.029978
## rm                  75.160001  1        8.669487
## age                 20.221256  1        4.496805
## dis                  3.098956  1        1.760385
## tax                172.940773  1       13.150695
## ptratio            128.350853  1       11.329204
## as.factor(chas)      1.106859  1        1.052074
## as.factor(rad)      19.817951  8        1.205220
## I(tlstat)            5.386236  1        2.320826
## I(tlstat^2)          2.435865  1        1.560726
## rm:tax             152.120733  1       12.333723
## rm:ptratio         144.537280  1       12.022366
## crim:nox           420.783738  1       20.513014
## nox:age             33.349899  1        5.774937
## crim:ptratio     12138.310514  1      110.174001
```

```
sqrt(vif(neww)) > 2 # problem?
```

```
##                    GVIF    Df GVIF^(1/(2*Df))
## crim               TRUE FALSE            TRUE
## rm                 TRUE FALSE            TRUE
## age                TRUE FALSE            TRUE
## dis               FALSE FALSE           FALSE
## tax                TRUE FALSE            TRUE
## ptratio            TRUE FALSE            TRUE
## as.factor(chas)   FALSE FALSE           FALSE
## as.factor(rad)     TRUE  TRUE           FALSE
## I(tlstat)          TRUE FALSE           FALSE
## I(tlstat^2)       FALSE FALSE           FALSE
## rm:tax             TRUE FALSE            TRUE
## rm:ptratio         TRUE FALSE            TRUE
## crim:nox           TRUE FALSE            TRUE
## nox:age            TRUE FALSE            TRUE
## crim:ptratio       TRUE FALSE            TRUE
```

As we can see from above there are some issues with multicolinearity. These issues may be due to having an R_k^2 of ≥ 0.9 or even just from interaction terms.