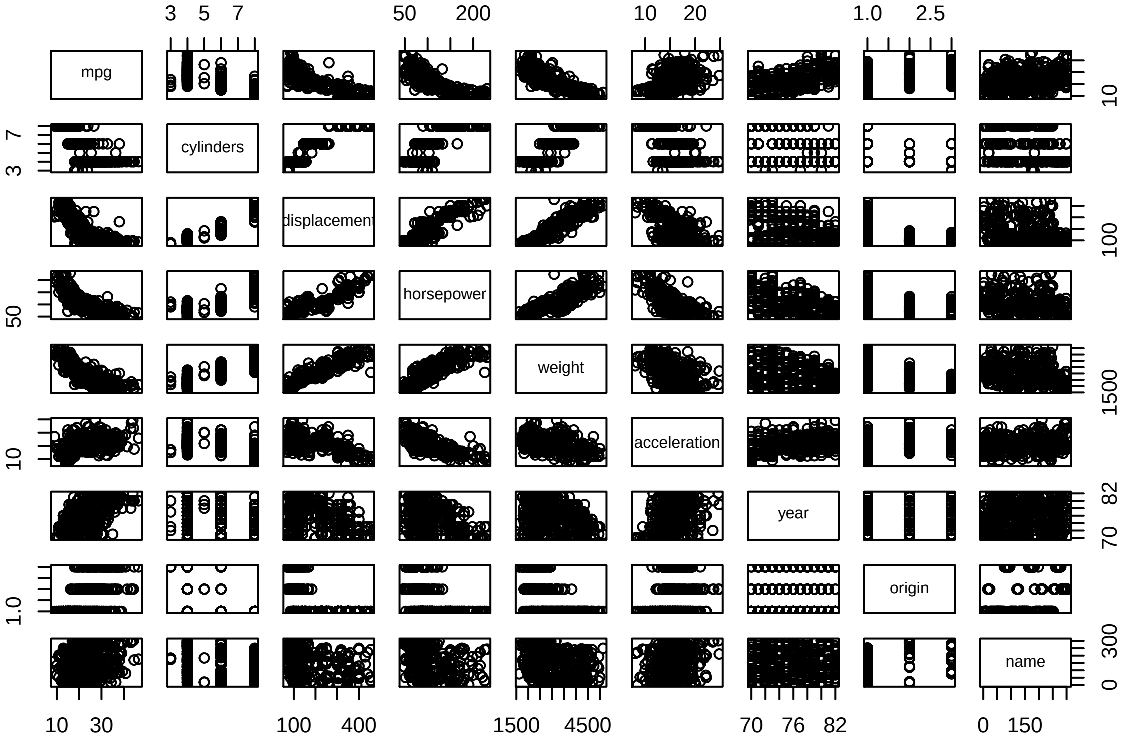
HW3 Solution CS6220-Data Mining

1. JWHT problem 9, p. 122

(a)

```
Auto = read.csv("/Users/ovitek/Dropbox/Olga/Teaching/CS6220/Fall15/Homeworks/Hw3/Auto.csv", header=T, na
Auto = na.omit(Auto)
pairs(Auto)
```



(b)

```
#corrplot(cor(Auto[,c(1:8)]), method = "number", type = "upper")
#cor(subset(Auto, select=-name))
cor(Auto[,c(1:8)])
```

```
##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442
## cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273
## displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944
## horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377
## weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000
## acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392
## year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199
## origin        0.5652088 -0.5689316   -0.6145351 -0.4551715 -0.5850054
##              acceleration       year     origin
## mpg             0.4233285  0.5805410  0.5652088
## cylinders      -0.5046834 -0.3456474 -0.5689316
## displacement   -0.5438005 -0.3698552 -0.6145351
## horsepower     -0.6891955 -0.4163615 -0.4551715
## weight         -0.4168392 -0.3091199 -0.5850054
## acceleration    1.0000000  0.2903161  0.2127458
## year            0.2903161  1.0000000  0.1815277
## origin          0.2127458  0.1815277  1.0000000
```

(c)

```r
fit1 = lm(mpg~.-name, data=Auto)
summary(fit1)
```

```
##
## Call:
## lm(formula = mpg ~ . - name, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5903 -2.1565 -0.1169  1.8690 13.0604
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
## cylinders     -0.493376   0.323282  -1.526  0.12780
## displacement   0.019896   0.007515   2.647  0.00844 **
## horsepower    -0.016951   0.013787  -1.230  0.21963
## weight        -0.006474   0.000652  -9.929  < 2e-16 ***
## acceleration   0.080576   0.098845   0.815  0.41548
## year           0.750773   0.050973  14.729  < 2e-16 ***
## origin         1.426141   0.278136   5.127 4.67e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.328 on 384 degrees of freedom
## Multiple R-squared:  0.8215, Adjusted R-squared:  0.8182
## F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

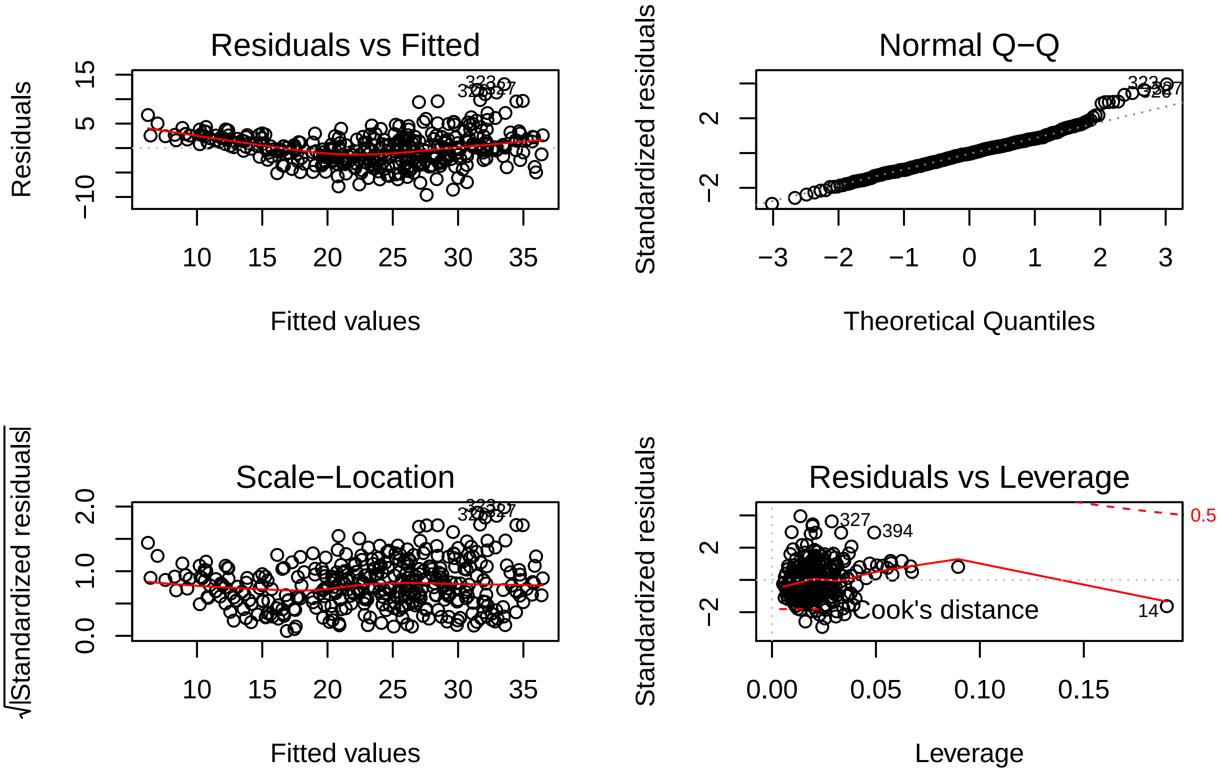
i. F is much larger than 1 and p-value is very small. So the null hupothesis is rejected and there is a relationship betwwn predictors and mpg.

ii. Displacement, weight, year, and origin have a small p-value and statistically significant relationship with mpg. On the other hand cylinders, horsepower, and acceleration do not.

iii. The coefficient of year is positive and approximately 0.75. it implies that every year the mpg is improved 75 %.

(d)

```r
par(mfrow=c(2,2))
plot(fit1)
```



The residual vs fitted plot is not a straight line, so, the assumptions of the model (specifically, non-systematic residuals with constant variance) are not verified. Therefore the model does not fit very well. Also, several points seems to have a large residual.

The leverage plot shows that point 14 is a high-leverage point

(e)

```
fit2 = lm(mpg~year*origin+displacement*weight+cylinders*year, data=Auto)
summary(fit2)
```

```
##
## Call:
## lm(formula = mpg ~ year * origin + displacement * weight + cylinders *
##     year, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.2316 -1.5732 -0.0506  1.3375 13.8294
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -3.521e+01  1.907e+01  -1.846  0.06564 .
## year                   1.164e+00  2.434e-01   4.782 2.48e-06 ***
## origin                -7.192e+00  4.909e+00  -1.465  0.14375
## displacement          -7.794e-02  1.083e-02  -7.195 3.32e-12 ***
## weight                -1.017e-02  6.604e-04 -15.393  < 2e-16 ***
## cylinders              7.945e+00  2.561e+00   3.102  0.00207 **
## year:origin            9.661e-02  6.314e-02   1.530  0.12684
## displacement:weight  1.988e-05  2.261e-06   8.795  < 2e-16 ***
## year:cylinders      -9.981e-02  3.271e-02  -3.052  0.00243 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.94 on 383 degrees of freedom
## Multiple R-squared:  0.861,  Adjusted R-squared:  0.8581
## F-statistic: 296.6 on 8 and 383 DF,  p-value: < 2.2e-16
```

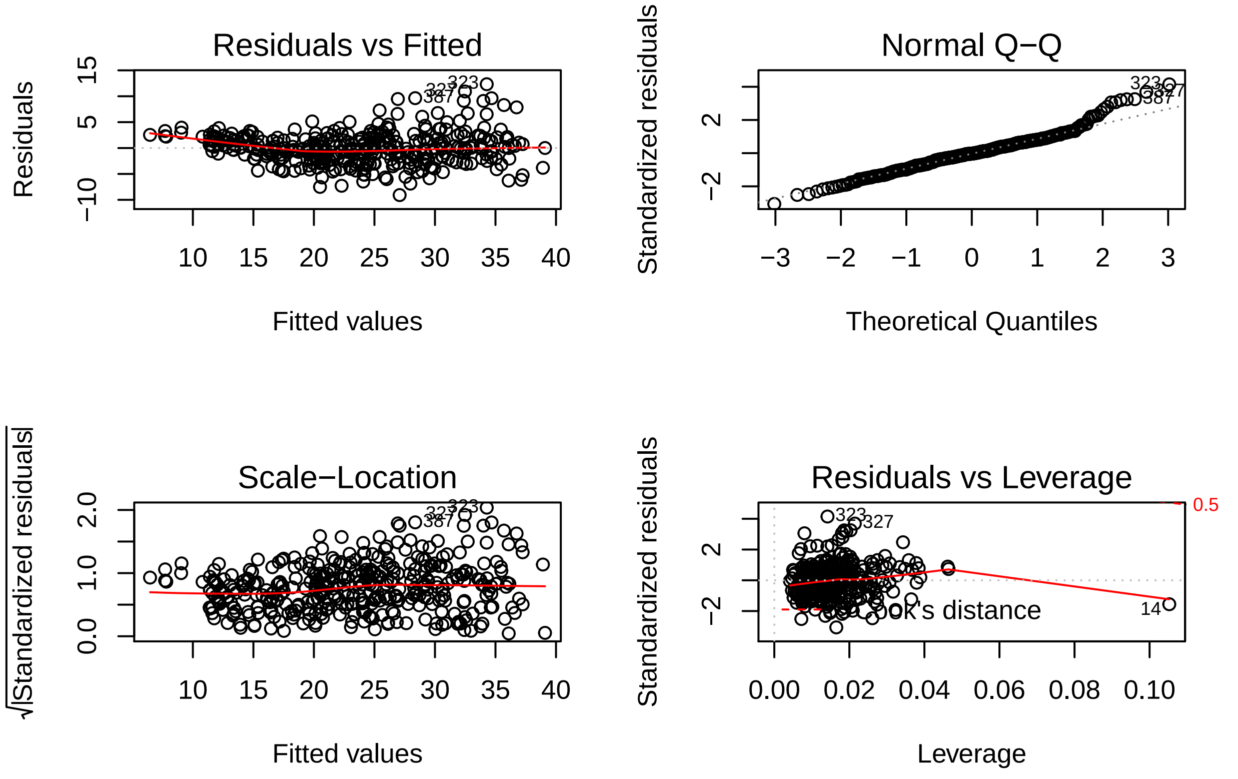
Checking the p-values show that the interaction of displacement and weight are significant. The other pairs are not that much. However, we need to keep in mind that some interaction termscan introduce multicillonearity, and this affects the standard errors of the parameter estimates, the p-values, and the resulting interpretation.

(f)

```r
fit3 = lm(mpg~log(weight)+sqrt(horsepower)+I(displacement^2)+I(year^2)+origin, data=Auto)
summary(fit3)
```

```
##
## Call:
## lm(formula = mpg ~ log(weight) + sqrt(horsepower) + I(displacement^2) +
##     I(year^2) + origin, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0832 -1.9044 -0.0867  1.7019 12.3210
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.548e+02  9.328e+00  16.595  < 2e-16 ***
## log(weight)       -1.942e+01  1.287e+00 -15.095  < 2e-16 ***
## sqrt(horsepower)  -1.013e+00  2.185e-01  -4.638 4.82e-06 ***
## I(displacement^2)  4.156e-05  7.763e-06   5.353 1.49e-07 ***
## I(year^2)          5.154e-03  3.013e-04  17.103  < 2e-16 ***
## origin             1.012e+00  2.421e-01   4.180 3.61e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.992 on 386 degrees of freedom
## Multiple R-squared:  0.8549, Adjusted R-squared:  0.853
## F-statistic: 454.9 on 5 and 386 DF,  p-value: < 2.2e-16
```

```r
par(mfrow=c(2,2))
plot(fit3)
```



The residual plot looks better. The F-statistics is larger now and all the predictors seem to be significant.

2. JWHT problem 13, p. 124 (a)

```r
set.seed(123)
x = rnorm(100)
```

(b)

```r
eps = rnorm(100, 0, sqrt(0.25))
```

(c)

```r
y = -1 + 0.5*x + eps
```

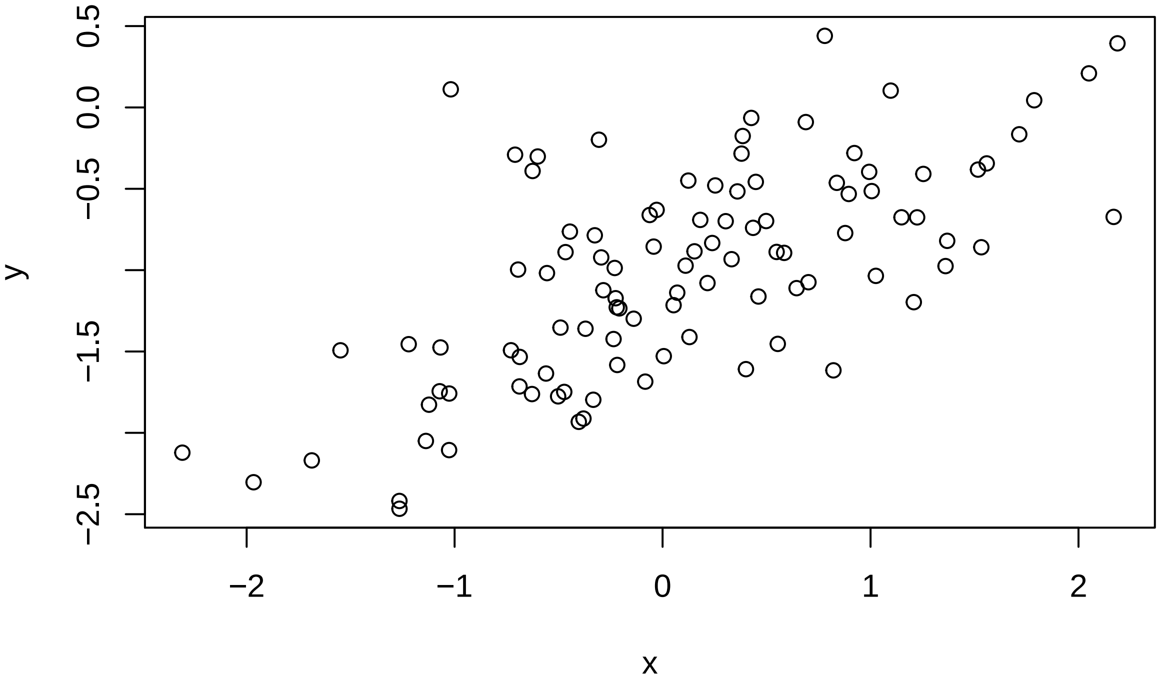
Length of y: 100

β_0: -1

β_1: 0.5

(d)

```r
plot(x, y)
```



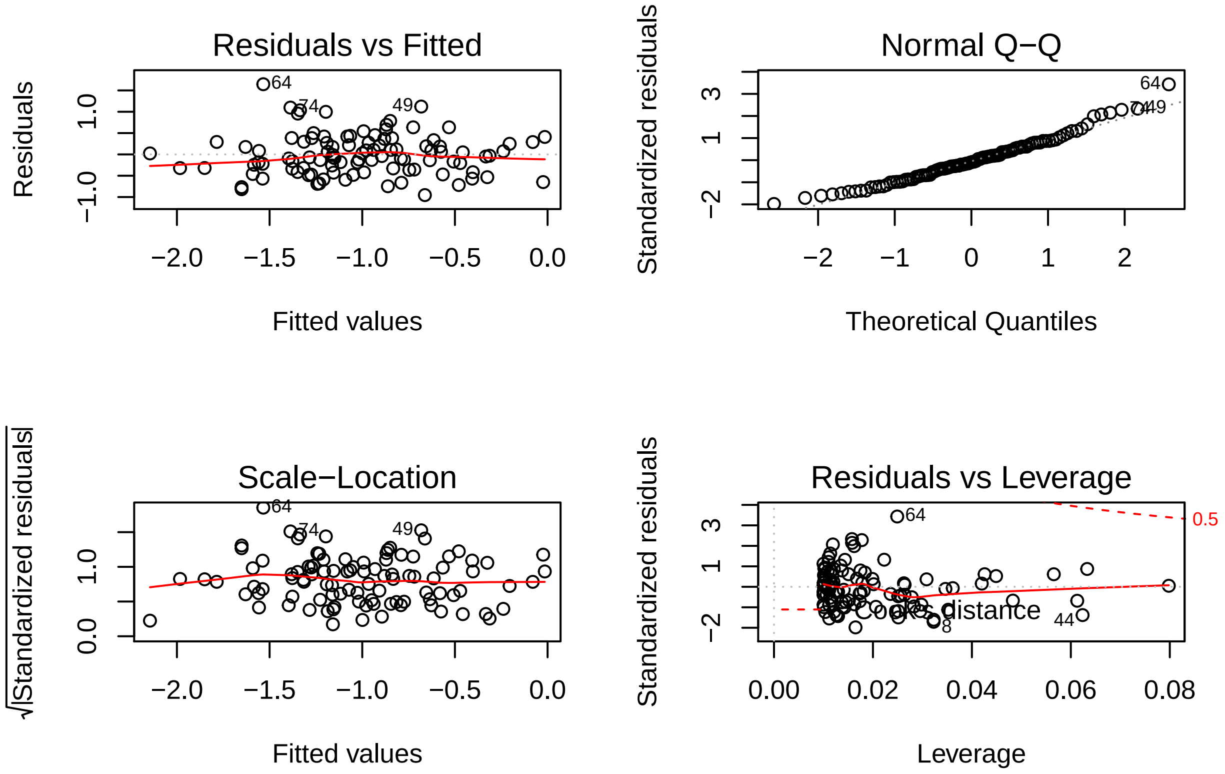
We see a positive linear relationship between y and x. The variance of the data was expected due to the noise included in the definition of y with respect to x.

(e)

```
fit1 = lm(y~x)
summary(fit1)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.95367 -0.34175 -0.04375  0.29032  1.64520
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.05140    0.04878 -21.556  < 2e-16 ***
## x            0.47376    0.05344   8.865  3.5e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4854 on 98 degrees of freedom
## Multiple R-squared:  0.4451, Adjusted R-squared:  0.4394
## F-statistic:  78.6 on 1 and 98 DF,  p-value: 3.497e-14
```

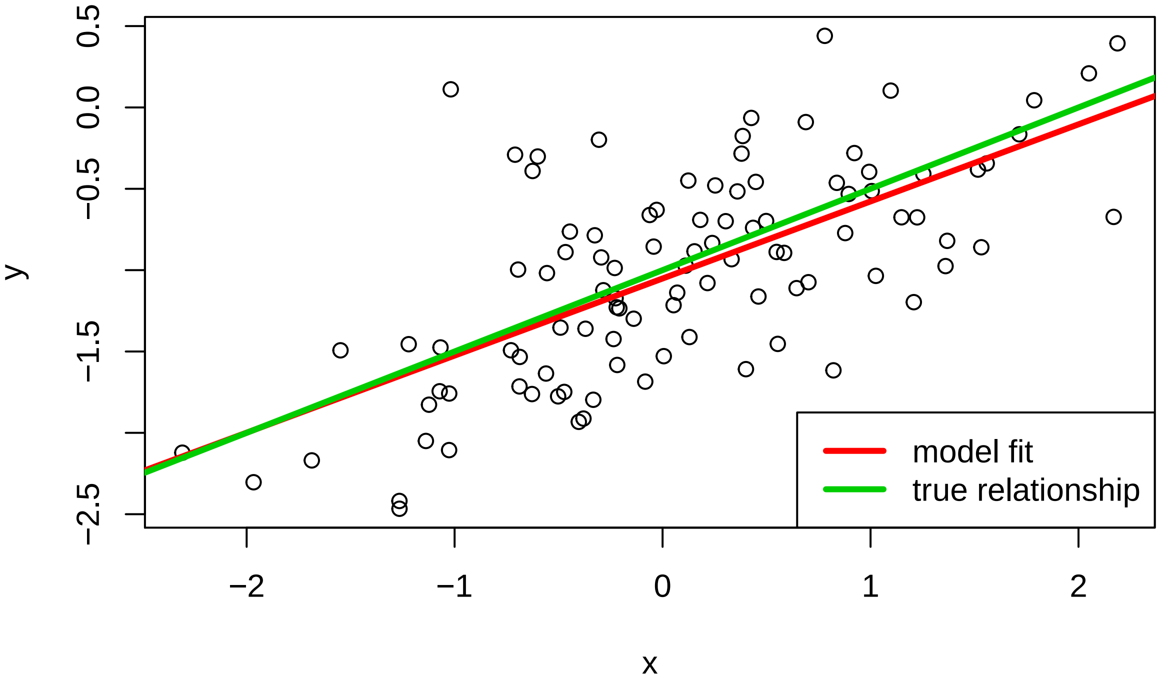
```
par(mfrow=c(2,2))
plot(fit1)
```



An advantage of the simulation is that we can compare the parameter values estimated from the data to the ground truth. Here $\hat{\beta}_0$ and $\hat{\beta}_1$ are very close to β_0 and β_1 respectively. The relatively large F-statistic and very small p-value reject the null hypothesis.

(f)

```r
plot(x, y)
abline(fit1, lwd=3, col=2)
abline(-1, 0.5, lwd=3, col=3)
legend('bottomright', legend = c("model fit", "true relationship"), lwd=3, col=2:3)
```



The fitted line is very close to the true line.

(g)

```
fit2 = lm(y~x+I(x^2))
summary(fit2)
```

```
##
## Call:
## lm(formula = y ~ x + I(x^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.96841 -0.36457 -0.05954  0.32720  1.66598
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.01520    0.06017 -16.873  < 2e-16 ***
## x            0.48428    0.05440   8.903 3.13e-14 ***
## I(x^2)      -0.04460    0.04342  -1.027    0.307
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4852 on 97 degrees of freedom
## Multiple R-squared:  0.451,  Adjusted R-squared:  0.4397
## F-statistic: 39.85 on 2 and 97 DF,  p-value: 2.333e-13
```

F-statistic decreased and the p-value of x^2 is large. So, there is no relationship between y and x^2. A polinomial regression is not justified by the data.

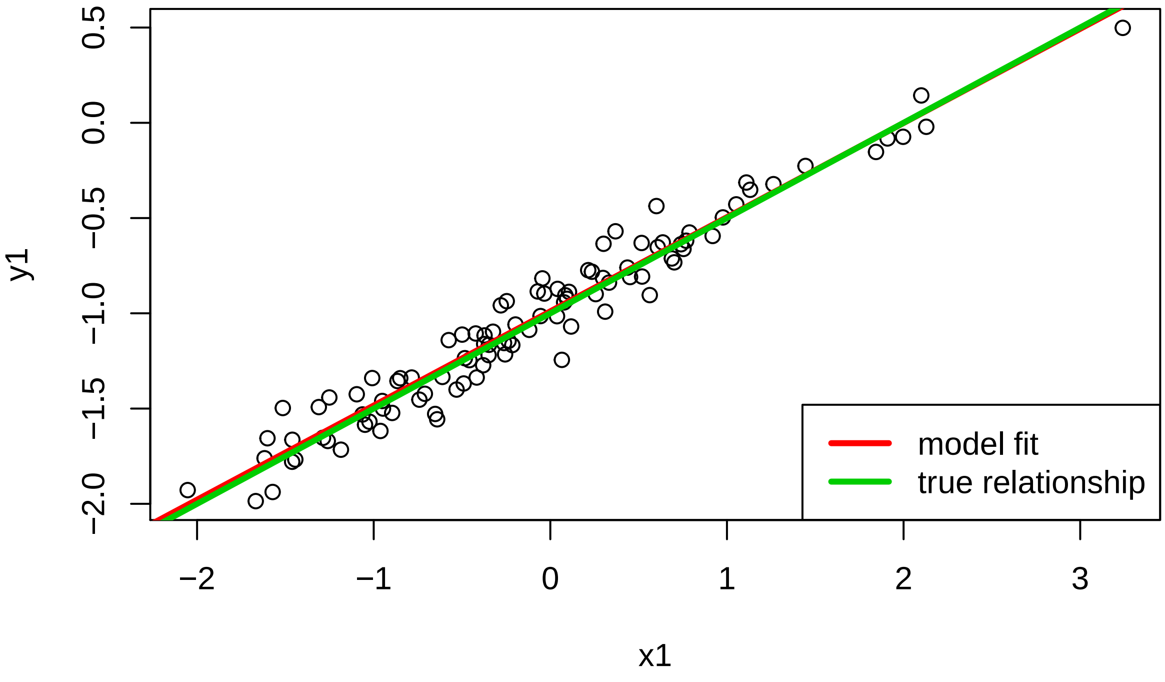
(h)

Less noise in the data means a smaller variance of the deviations around the straight line.

```
set.seed(123)
eps1 = rnorm(100, 0, 0.12)
x1 = rnorm(100)
y1 = -1 + 0.5*x1 + eps1
fit2 = lm(y1~x1)
summary(fit2)
```

```
##
## Call:
## lm(formula = y1 ~ x1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.286979 -0.071484 -0.005167  0.071068  0.255605
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.98975    0.01106  -89.45   <2e-16 ***
## x1           0.49439    0.01143   43.26   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.11 on 98 degrees of freedom
## Multiple R-squared:  0.9502, Adjusted R-squared:  0.9497
## F-statistic:  1871 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
plot(x1, y1)
abline(fit2, lwd=3, col=2)
abline(-1, 0.5, lwd=3, col=3)
legend('bottomright', legend = c("model fit", "true relationship"), lwd=3, col=2:3)
```



The estimated parameters are closer to the truth, the fitted line is closer to the straight line, and residual standard error decreased as expected.

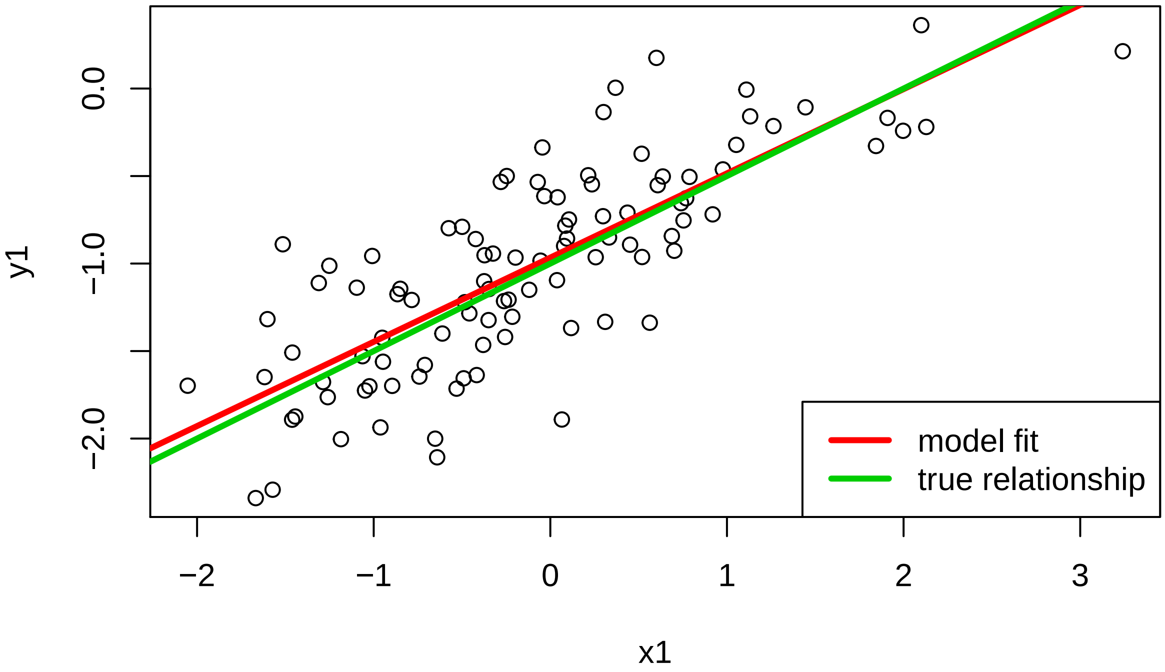
(i)

More noise means a larger variance of deviations from the straight line.

```
set.seed(123)
eps1 = rnorm(100, 0, 0.4)
x1 = rnorm(100)
y1 = -1 + 0.5*x1 + eps1
fit3 = lm(y1~x1)
summary(fit3)
```

```
##
## Call:
## lm(formula = y1 ~ x1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.95660 -0.23828 -0.01722  0.23689  0.85202
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.96585    0.03688  -26.19   <2e-16 ***
## x1           0.48130    0.03810   12.63   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3665 on 98 degrees of freedom
## Multiple R-squared:  0.6196, Adjusted R-squared:  0.6157
## F-statistic: 159.6 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
plot(x1, y1)
abline(fit3, lwd=3, col=2)
abline(-1, 0.5, lwd=3, col=3)
legend('bottomright', legend = c("model fit", "true relationship"), lwd=3, col=2:3)
```



The Residual standard error increased as expected, and there is a somewhat greater difference between the true and the fitted line.

(j)

```
confint(fit1)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.1481951 -0.9546080
## x            0.3677156  0.5798128
```

```
confint(fit2)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.0117119 -0.9677976
## x1           0.4717092  0.5170690
```

```
confint(fit3)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.0390396 -0.8926586
## x1           0.4056973  0.5568968
```

The less noisy data has narrower and the noisier data has wider confidence intervals compare to the one of the original data.

3. JWHT problem 15, p. 126 (a)

```
library(MASS)
summary(Boston)
```

```
##       crim                zn             indus            chas        
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000  
##  1st Qu.: 0.08204   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000  
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000  
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917  
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000  
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000  
##       nox               rm             age              dis        
##  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130  
##  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100  
##  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207  
##  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795  
##  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188  
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127  
##       rad              tax           ptratio          black       
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32  
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38  
##  Median : 5.000   Median :330.0   Median :19.05   Median :391.44  
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67  
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23  
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90  
##      lstat            medv      
##  Min.   : 1.73   Min.   : 5.00  
##  1st Qu.: 6.95   1st Qu.:17.02  
##  Median :11.36   Median :21.20  
##  Mean   :12.65   Mean   :22.53  
##  3rd Qu.:16.95   3rd Qu.:25.00  
##  Max.   :37.97   Max.   :50.00
```

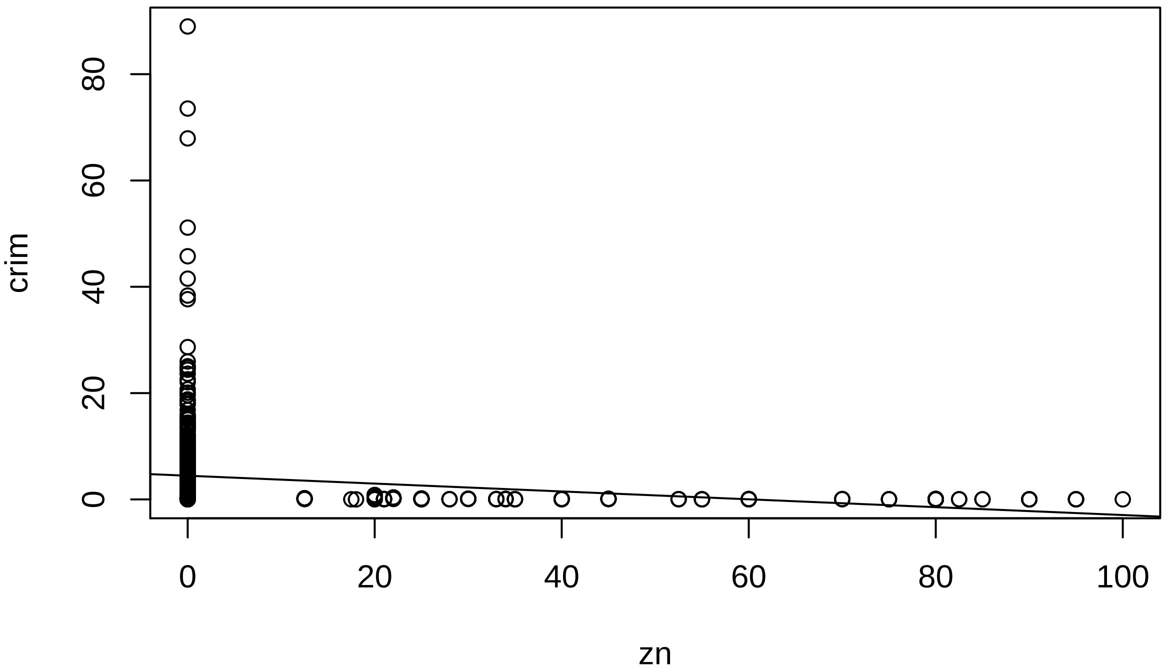
```
attach(Boston)
```

For each predictor, you should run the following to check the lm result for crime vs each predictor.

```
par(mfrow=c(1,1))
lm.zn = lm(crim ~ zn)
summary(lm.zn)
```

```
## 
## Call:
## lm(formula = crim ~ zn)
## 
## 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
```

```
plot(crim ~ zn)
abline(lm.zn)
```



```
# Repeat the same for the other predictors
```

The models with only one predictor generally fit very poorly

(b)

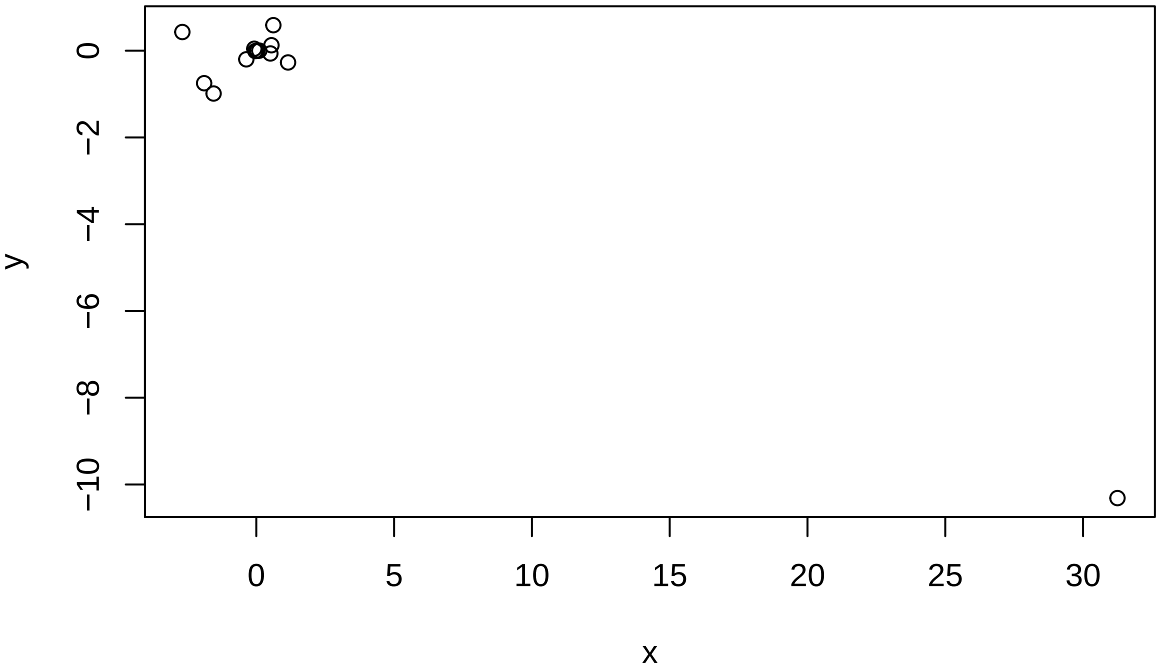
```
lm.all = lm(crim~., data=Boston)
summary(lm.all)
```

```
##
## Call:
## lm(formula = crim ~ ., data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.924 -2.120 -0.353  1.019 75.051
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.033228   7.234903   2.354 0.018949 *
## zn            0.044855   0.018734   2.394 0.017025 *
## indus        -0.063855   0.083407  -0.766 0.444294
## chas         -0.749134   1.180147  -0.635 0.525867
## nox         -10.313535   5.275536  -1.955 0.051152 .
## rm            0.430131   0.612830   0.702 0.483089
## age           0.001452   0.017925   0.081 0.935488
## dis          -0.987176   0.281817  -3.503 0.000502 ***
## rad           0.588209   0.088049   6.680 6.46e-11 ***
## tax          -0.003780   0.005156  -0.733 0.463793
## ptratio      -0.271081   0.186450  -1.454 0.146611
## black        -0.007538   0.003673  -2.052 0.040702 *
## lstat         0.126211   0.075725   1.667 0.096208 .
## medv         -0.198887   0.060516  -3.287 0.001087 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.439 on 492 degrees of freedom
## Multiple R-squared:  0.454,  Adjusted R-squared:  0.4396
## F-statistic: 31.47 on 13 and 492 DF,  p-value: < 2.2e-16
```

These predictors are statistically significant: zn, dis, rad, black, medv. However we should look beyond statistical significance, and investigate the issues of multicollinearity, violation of model assumptions, outliers etc.

(c)

```
x <- c(coefficients(lm(crim ~ zn, data=Boston))[2],
      coefficients(lm(crim ~ indus, data=Boston))[2],
      coefficients(lm(crim ~ chas, data=Boston))[2],
      coefficients(lm(crim ~ nox, data=Boston))[2],
      coefficients(lm(crim ~ rm, data=Boston))[2],
      coefficients(lm(crim ~ age, data=Boston))[2],
      coefficients(lm(crim ~ dis, data=Boston))[2],
      coefficients(lm(crim ~ rad, data=Boston))[2],
      coefficients(lm(crim ~ tax, data=Boston))[2],
      coefficients(lm(crim ~ ptratio, data=Boston))[2],
      coefficients(lm(crim ~ black, data=Boston))[2],
      coefficients(lm(crim ~ lstat, data=Boston))[2],
      coefficients(lm(crim ~ medv, data=Boston))[2])
y = coefficients(lm.all)[2:14]
plot(x, y)
```



While the majority of the coefficients did not change, the coefficient of nox changed substantially after uncluding the other variables.

(d) For example, for zn:

```
lm.zn = lm(crim~poly(zn,3))
summary(lm.zn)
```

```
##
## Call:
## lm(formula = crim ~ poly(zn, 3))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.821 -4.614 -1.294  0.473 84.130
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.6135     0.3722   9.709  < 2e-16 ***
## poly(zn, 3)1 -38.7498     8.3722  -4.628  4.7e-06 ***
## poly(zn, 3)2  23.9398     8.3722   2.859  0.00442 **
## poly(zn, 3)3 -10.0719     8.3722  -1.203  0.22954
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.372 on 502 degrees of freedom
## Multiple R-squared:  0.05824,    Adjusted R-squared:  0.05261
## F-statistic: 10.35 on 3 and 502 DF,  p-value: 1.281e-06
```

For examplefor zn, and zn^2 are statistically significant but zn^3 is not. You may see except for black and chas, other parameters have some nonlinear relation to crime.