HW2

Yigao Li

February 1, 2018

3.7 - 4

(a)

Residual Sum of Squares (RSS) for cubic regression must be less than linear regression because of least squares method while creating regression models.

(b)

However, when dealing with test data, RSS for cubic regression can be larger than linear one since the true relationship between X and Y is linear.

(c)

RSS for cubic regression should be less than linear regression because higher order polynomial has more flexibility.

(d)

There is not enough information to tell which one has a smaller RSS because we don't know the real relationship between X and Y.

3.7 - 9

(a)

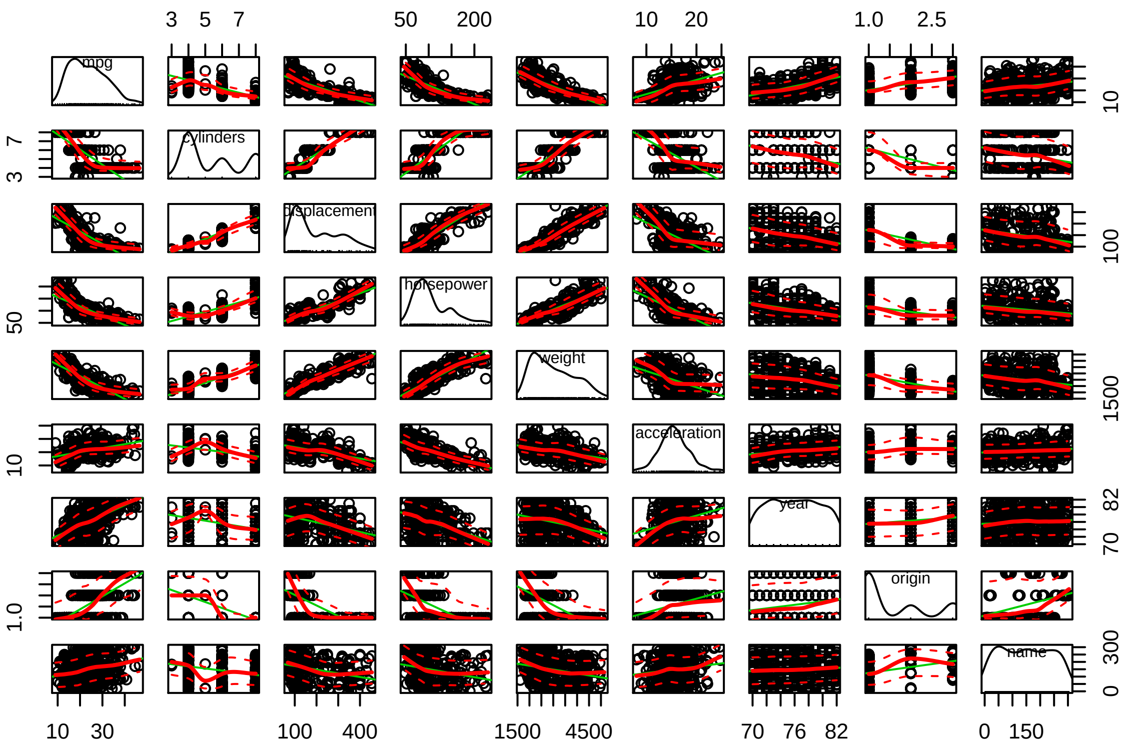
```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 3.4.3
```

```
library(car)
```

```
## Warning: package 'car' was built under R version 3.4.3
```

```
auto <- Auto
scatterplotMatrix(auto)
```



(b)

```r
auto.cor <- subset(auto, select = -name)
cor(auto.cor)
```

```
##                     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
model.1 <- lm(mpg ~ ., data = auto.cor)
summary(model.1)
```

```
##
## Call:
## lm(formula = mpg ~ ., data = auto.cor)
##
## 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.

There is a relationship between mpg and other variables because the F statistics is large.

ii.

"displacement", "weight", "year", "origin" have statistically significant relationship to "mpg"

iii.

Keep other variables unchanged, the estimated mpg of a car increases 0.750773 every year.

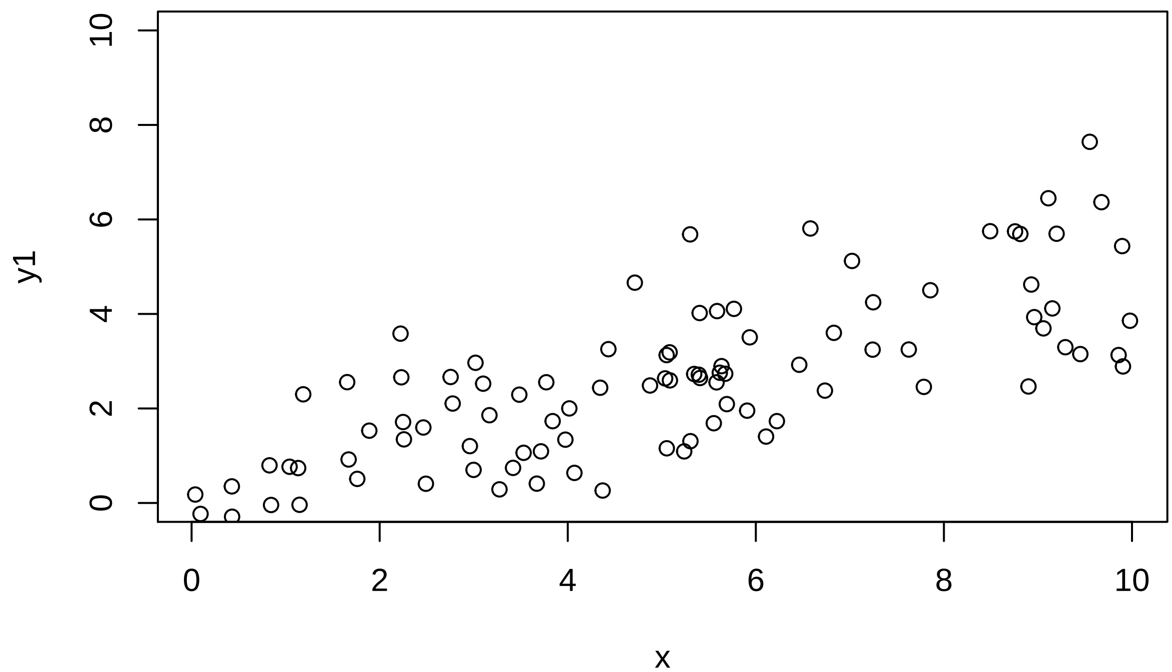
3.7 - 12

(a)

Let $y_i = \hat{\beta}x_i$ and $x_i = \hat{\alpha}y_i$. $\hat{\beta} = \frac{\sum x_i y_i}{\sum x_i^2}$, $\hat{\alpha} = \frac{\sum x_i y_i}{\sum y_i^2}$.
Therefore, to make $\hat{\beta} = \hat{\alpha}$, $\sum x_i^2 = \sum y_i^2$

(b)

```r
n <- 100
x <- sort(runif(100,0,10))    # x is random number from 0 to 5
y1 <- rnorm(n, x/2, 1)       # The true relation is y=2
plot(x,y1, ylim = c(0,10))
```




```r
model.2 <- lm(y1~x-1)        # Y to X
summary(model.2)             # coefficient is approximately 2
```

```
##
## Call:
## lm(formula = y1 ~ x - 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.66122 -0.83980  0.02996  0.62501  3.03058
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## x  0.50057    0.02116   23.66   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.191 on 99 degrees of freedom
## Multiple R-squared:  0.8497, Adjusted R-squared:  0.8482
```

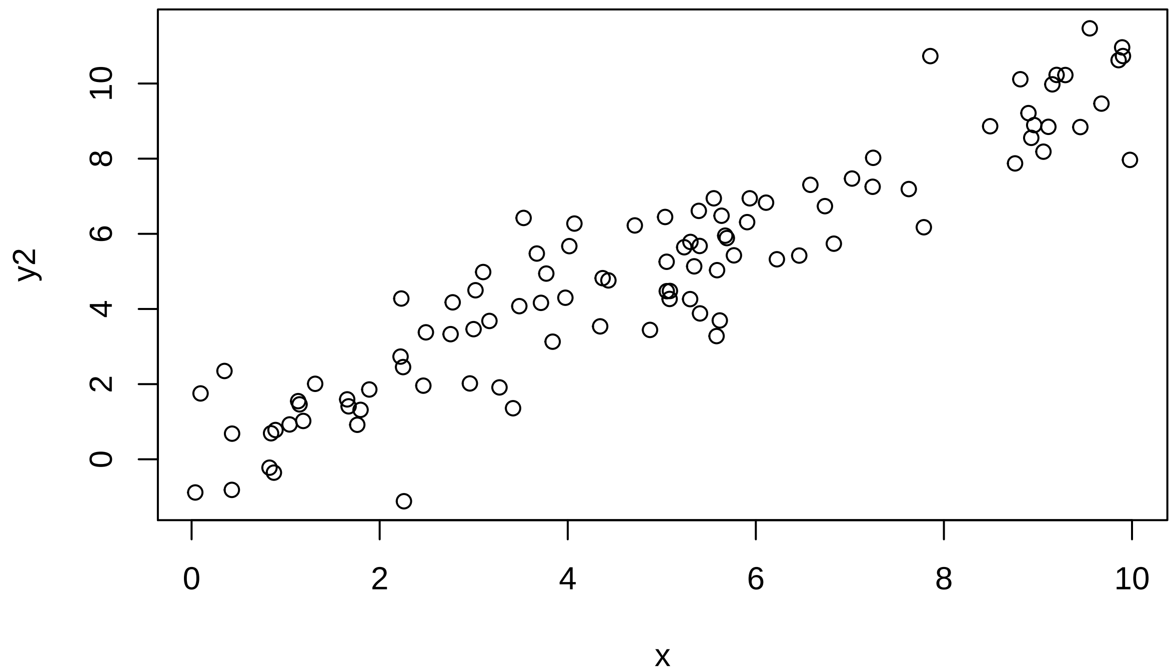
```
## F-statistic: 559.6 on 1 and 99 DF,  p-value: < 2.2e-16
```

```r
model.3 <- lm(x~y1-1)       # X to Y
summary(model.3)            # coefficient is approximately 0.5
```

```
##
## Call:
## lm(formula = x ~ y1 - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3475 -0.5333  0.7193  2.1932  4.9996
##
## Coefficients:
##    Estimate Std. Error t value Pr(>|t|)
## y1  1.69746    0.07175   23.66   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.193 on 99 degrees of freedom
## Multiple R-squared:  0.8497, Adjusted R-squared:  0.8482
## F-statistic: 559.6 on 1 and 99 DF,  p-value: < 2.2e-16
```

(c)

```r
y2 <- rnorm(n, x, 1)
plot(x,y2)
```




```r
model.4 <- lm(y2~x-1)
summary(model.4)
```

```
##
## Call:
## lm(formula = y2 ~ x - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4267 -0.7434  0.1476  0.6258  2.8102
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## x  1.02317    0.02008   50.96   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.13 on 99 degrees of freedom
## Multiple R-squared:  0.9633, Adjusted R-squared:  0.9629
## F-statistic:  2597 on 1 and 99 DF,  p-value: < 2.2e-16
```

```r
model.5 <- lm(x~y2-1)
summary(model.5)
```

```
##
## Call:
## lm(formula = x ~ y2 - 1)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5161 -0.3590  0.0869  0.8719  3.3090
##
## Coefficients:
##    Estimate Std. Error t value Pr(>|t|)
## y2  0.94147    0.01847   50.96   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.084 on 99 degrees of freedom
## Multiple R-squared:  0.9633, Adjusted R-squared:  0.9629
## F-statistic:  2597 on 1 and 99 DF,  p-value: < 2.2e-16
```

3.7 - 13

```r
set.seed(1)
```

(a)

```r
x <- rnorm(100)
```

(b)

```r
eps <- rnorm(100, 0, sqrt(0.25))
```

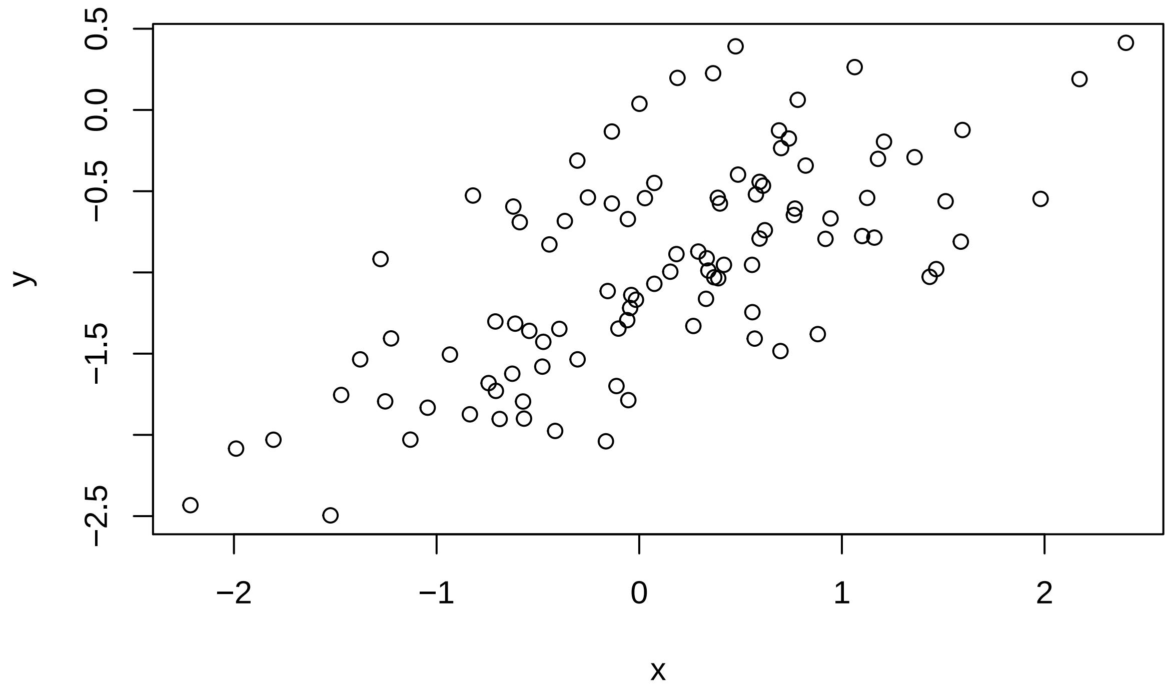
(c)

```r
y <- -1 + x/2 + eps
```

The length of y is 100. $\beta_0 = -1$. $\beta_1 = 0.5$

(d)

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



x and y show positive linear relation.

(e)

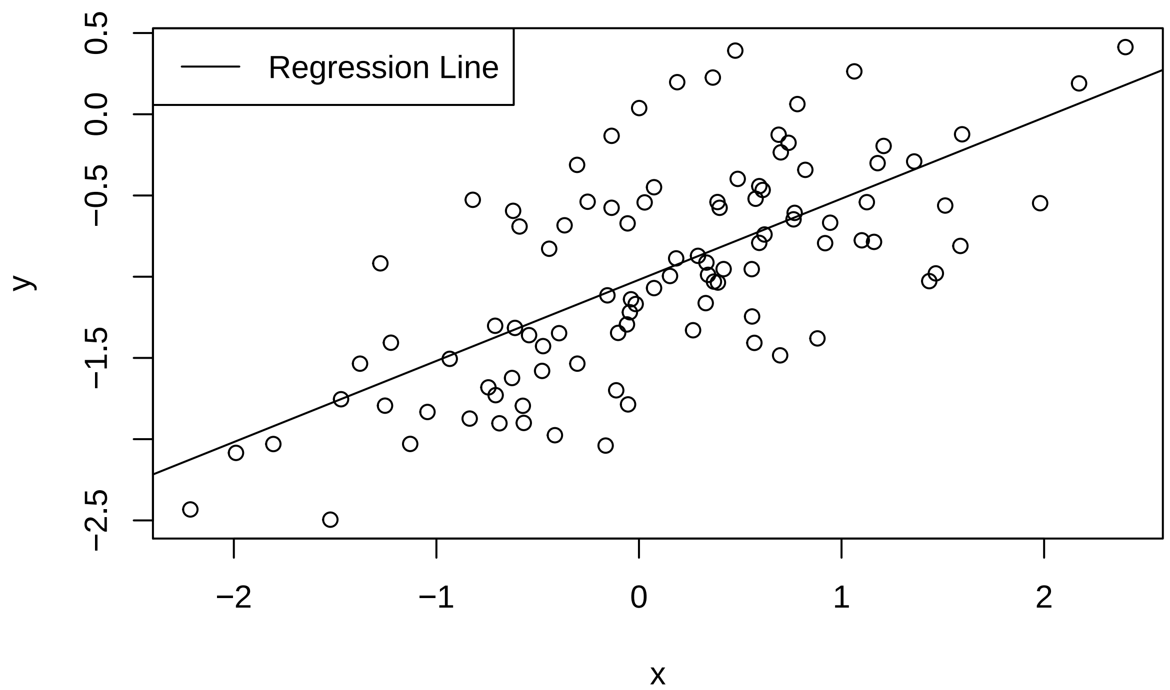
```
model.6 <- lm(y~x)
summary(model.6)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.93842 -0.30688 -0.06975  0.26970  1.17309
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.01885    0.04849 -21.010  < 2e-16 ***
## x            0.49947    0.05386   9.273 4.58e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4814 on 98 degrees of freedom
## Multiple R-squared:  0.4674, Adjusted R-squared:  0.4619
## F-statistic: 85.99 on 1 and 98 DF,  p-value: 4.583e-15
```

$\hat{\beta}_0 = -1.00478$, $\hat{\beta}_1 = 0.44156$. The estimated $\hat{\beta}$ values are approximately the same with real β value.

(f)

```r
plot(x,y)
abline(model.6)
legend("topleft", legend = "Regression Line", lty = 1)
```



(g)

```r
model.7 <- lm(y~x+I(x^2))
summary(model.7)
```

```
##
## Call:
## lm(formula = y ~ x + I(x^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.98252 -0.31270 -0.06441  0.29014  1.13500
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.97164    0.05883 -16.517  < 2e-16 ***
```

```
## x            0.50858    0.05399   9.420  2.4e-15 ***
## I(x^2)      -0.05946    0.04238  -1.403    0.164
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.479 on 97 degrees of freedom
## Multiple R-squared:  0.4779, Adjusted R-squared:  0.4672
## F-statistic:  44.4 on 2 and 97 DF,  p-value: 2.038e-14
```

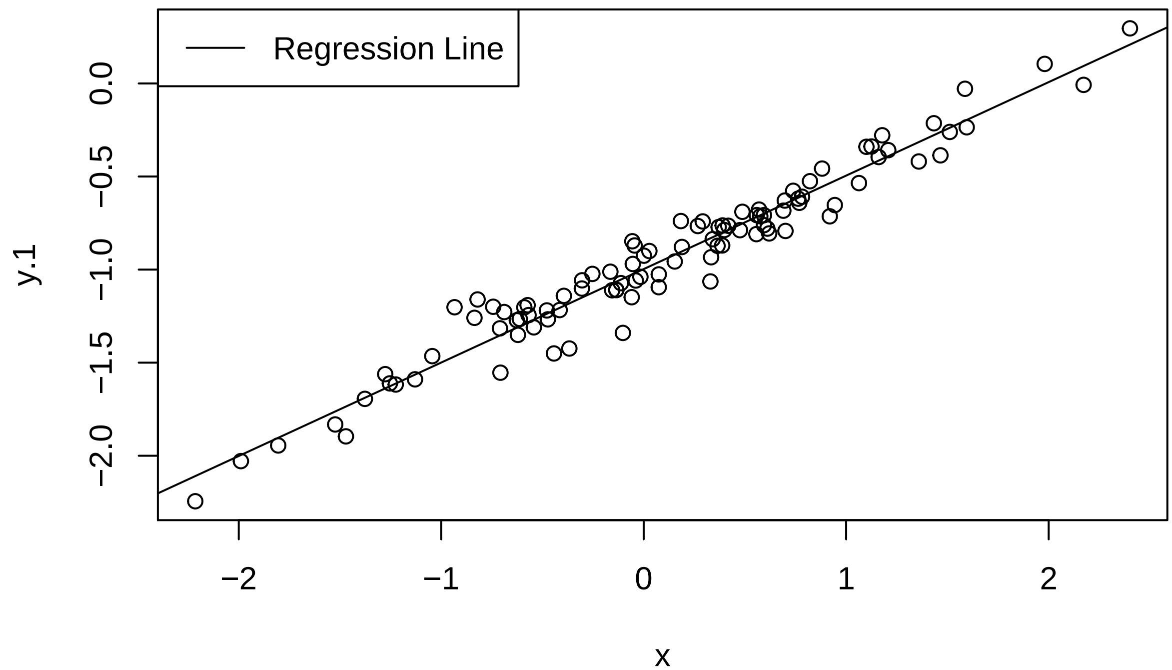
Although this polynomial regression model is also significant due to large F statistics, the p-value of coefficient of X^2 term is 0.589, meaning that adding quadratic term does not improve the model. Furthermore, adjusted R-squared is less than multiple R-squared, which also explains why we should not add x^2 term.

(h)

```r
eps.1 <- rnorm(100, 0, 0.1)
y.1 <- -1 + x/2 + eps.1
model.8 <- lm(y.1~x)
summary(model.8)
```

```
##
## Call:
## lm(formula = y.1 ~ x)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.291411 -0.048230 -0.004533  0.064924  0.264157
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.99726    0.01047  -95.25   <2e-16 ***
## x            0.50212    0.01163   43.17   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1039 on 98 degrees of freedom
## Multiple R-squared:  0.9501, Adjusted R-squared:  0.9495
## F-statistic:  1864 on 1 and 98 DF,  p-value: < 2.2e-16
```

```r
plot(x,y.1)
abline(model.8)
legend("topleft", legend = "Regression Line", lty = 1)
```



Variance of *eps* decreases to 0.01. The regression line is approximately the same but the new model has higher Multiple R-squared.

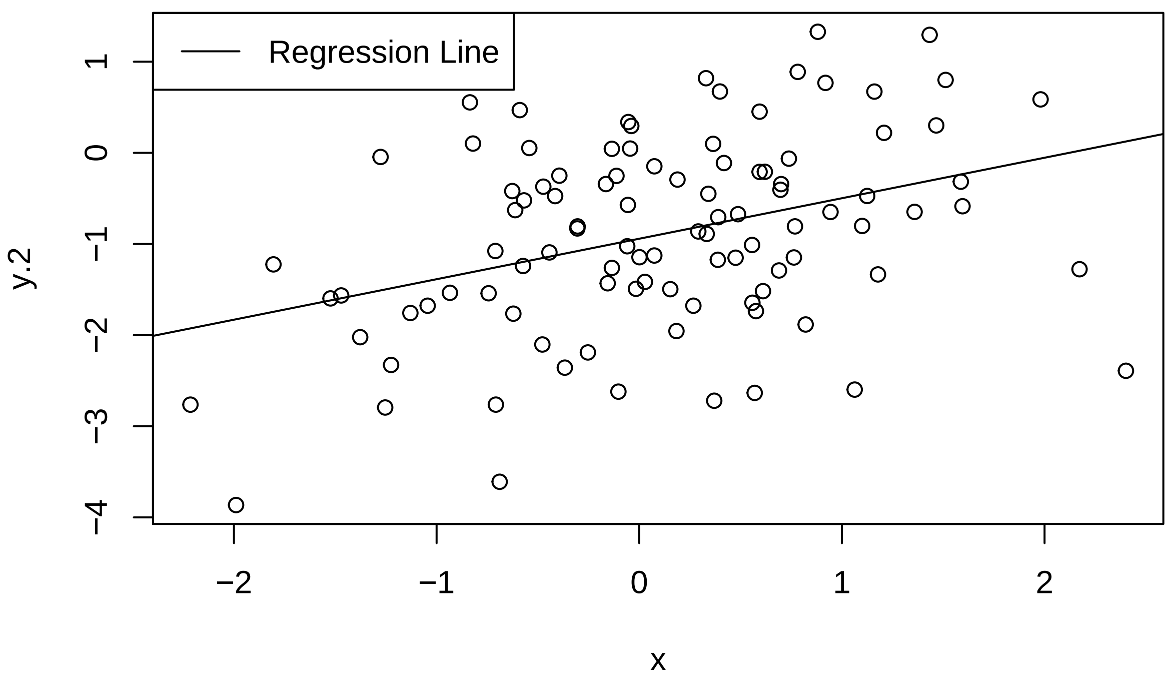
(i)

```
eps.2 <- rnorm(100, 0, 1)
y.2 <- -1 + x/2 + eps.2
model.9 <- lm(y.2~x)
summary(model.9)
```

```
##
## Call:
## lm(formula = y.2 ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.51626 -0.54525 -0.03776  0.67289  1.87887
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.9423     0.1003  -9.397 2.47e-15 ***
## x             0.4443     0.1114   3.989 0.000128 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.9955 on 98 degrees of freedom
## Multiple R-squared:  0.1397, Adjusted R-squared:  0.1309
## F-statistic: 15.91 on 1 and 98 DF,  p-value: 0.000128
```

```r
plot(x,y.2)
abline(model.9)
legend("topleft", legend = "Regression Line", lty = 1)
```



Variance of error increases to 1. The coefficient of term x moves away from 0.5 due to more noise.

(j)

```r
print("95% confidence interval of coefficients in original data")
```

```
## [1] "95% confidence interval of coefficients in original data"
```

```r
confint(model.6)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.1150804 -0.9226122
## x            0.3925794  0.6063602
```

```r
print("95% confidence interval of coefficients in less noise data")
```

```
## [1] "95% confidence interval of coefficients in less noise data"
```

```r
confint(model.8)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.0180413 -0.9764850
## x            0.4790377  0.5251957
```

```r
print("95% confidence interval of coefficients in more noise data")
```

```
## [1] "95% confidence interval of coefficients in more noise data"
```

```r
confint(model.9)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.1413399 -0.7433293
## x            0.2232721  0.6653558
```

If data points separate away from each other, confidence intervals of coefficients will have wider ranges.

3.7 - 15

(a)

```r
library(MASS)
boston <- Boston
x <- c()
for (predictor in names(boston)[-1]){
  simple.model <- paste("crim ~", predictor, sep = " ")
  column <- paste("boston$", predictor, sep = "")
  model.10 <- lm(simple.model, data = boston)
  x <- c(x, as.numeric(coef(model.10)[2]))
  print(summary(model.10))
}
```

```
##
## Call:
## lm(formula = simple.model, 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
##
##
## Call:
## lm(formula = simple.model, 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
##
##
## Call:
## lm(formula = simple.model, 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
##
##
## Call:
## lm(formula = simple.model, 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
##
##
```

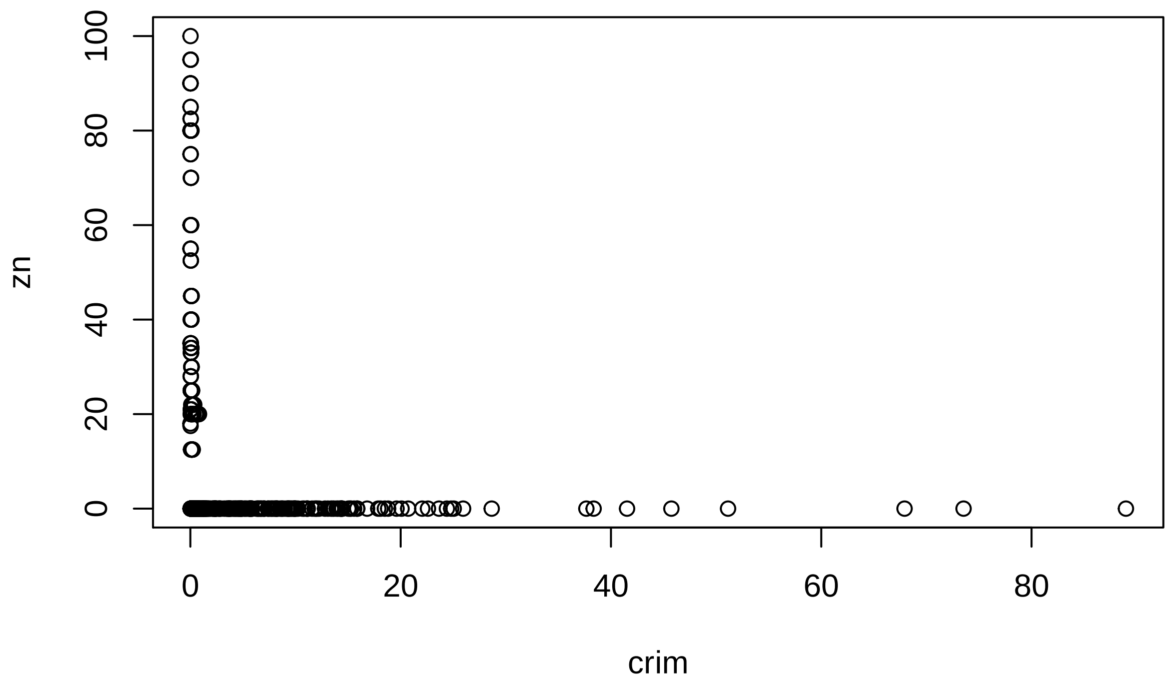
```
## Call:
## lm(formula = simple.model, data = boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.604 -3.952 -2.654  0.989 87.197
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   20.482      3.365   6.088 2.27e-09 ***
## rm            -2.684      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
##
##
## Call:
## lm(formula = simple.model, 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
##
##
## Call:
## lm(formula = simple.model, 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
```

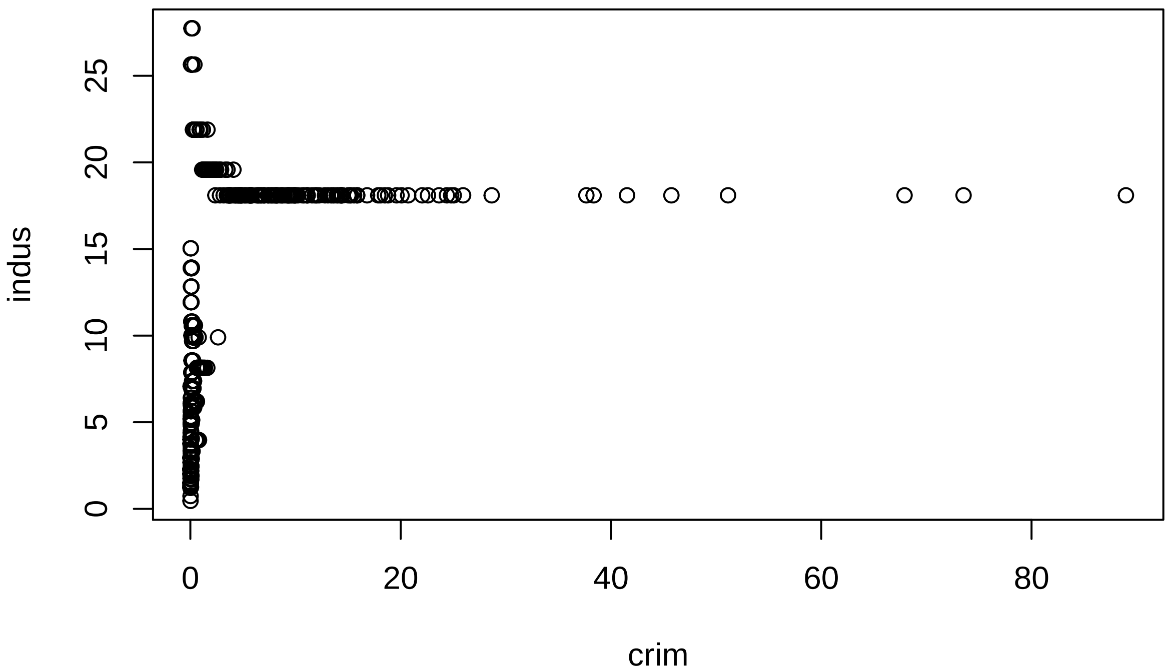
```
## F-statistic: 84.89 on 1 and 504 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = simple.model, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.164  -1.381  -0.141   0.660  76.433
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (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
##
##
## Call:
## lm(formula = simple.model, 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
##
##
## Call:
## lm(formula = simple.model, 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
```

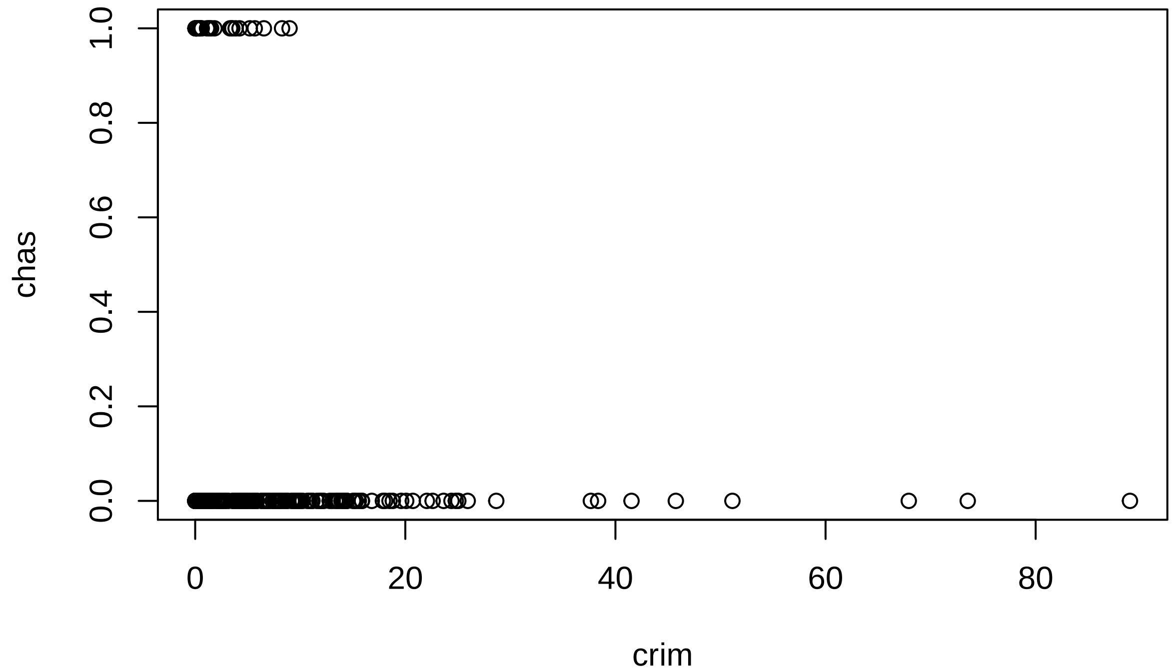
```
##
## Residual standard error: 8.24 on 504 degrees of freedom
## Multiple R-squared:  0.08407,    Adjusted R-squared:  0.08225
## F-statistic: 46.26 on 1 and 504 DF,  p-value: 2.943e-11
##
##
## Call:
## lm(formula = simple.model, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     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
##
##
## Call:
## lm(formula = simple.model, 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
##
##
## Call:
## lm(formula = simple.model, data = boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.071 -4.022 -2.343  1.298 80.957
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.79654    0.93419   12.63   <2e-16 ***
```

```
## medv        -0.36316    0.03839   -9.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.934 on 504 degrees of freedom
## Multiple R-squared:  0.1508, Adjusted R-squared:  0.1491
## F-statistic: 89.49 on 1 and 504 DF,  p-value: < 2.2e-16
```

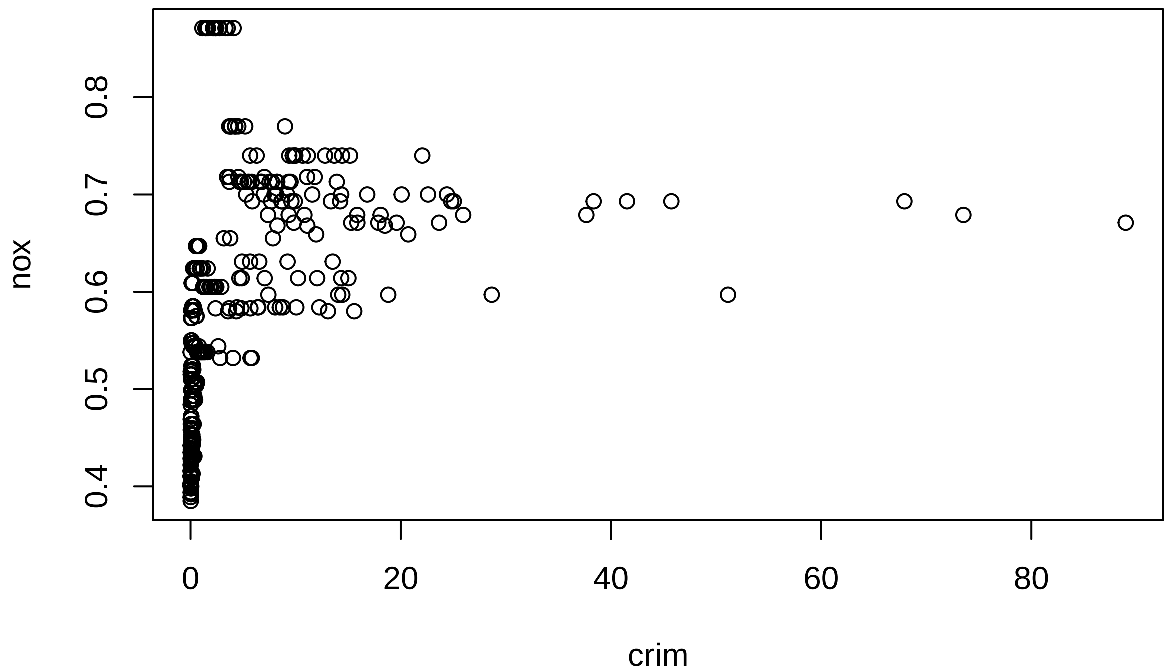
```r
for (i in 2:14){
  plot(boston[,c(1,i)])
}
```



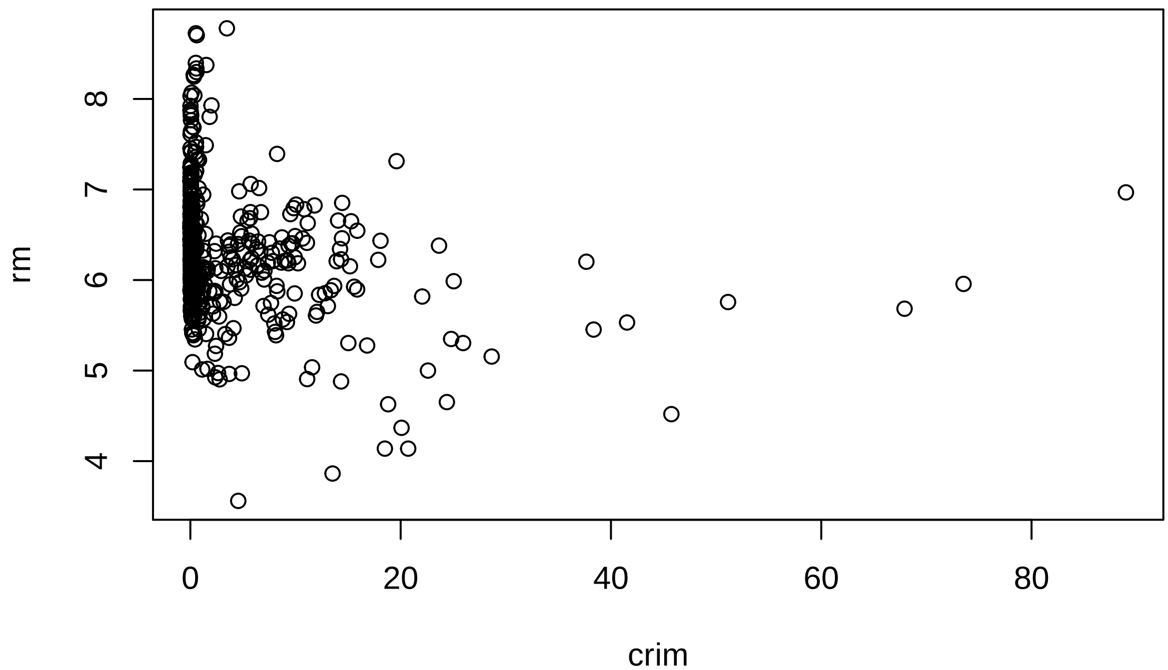

indus
0
5
10
15
20
25
0
20
40
60
80
crim

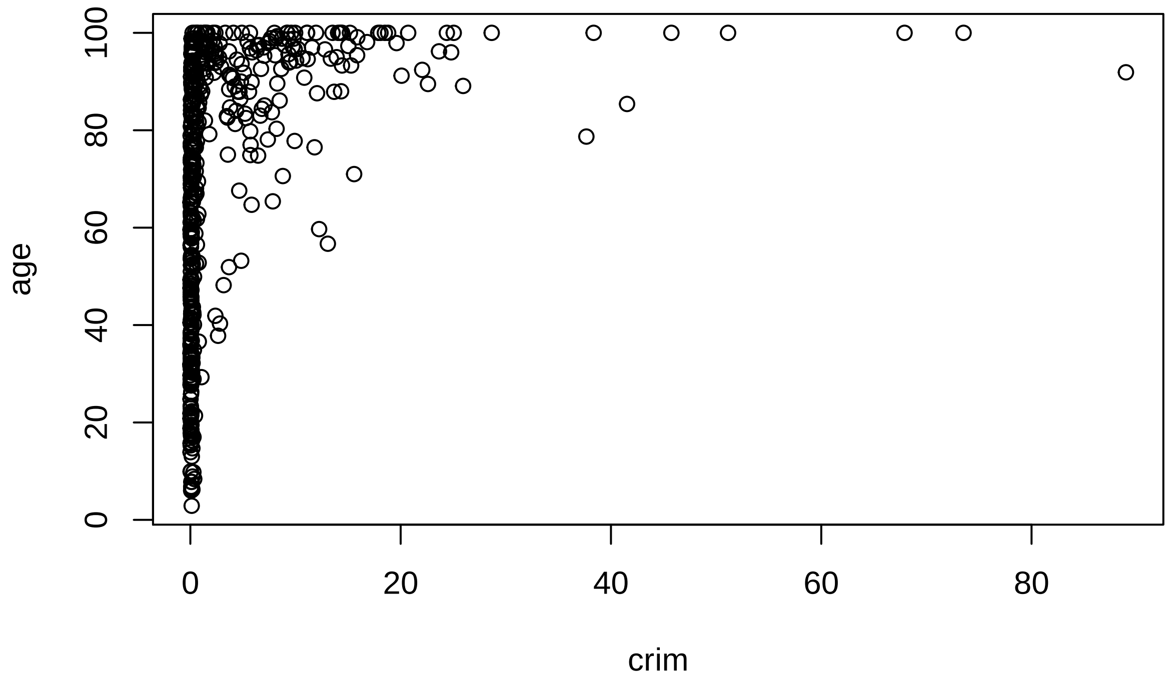


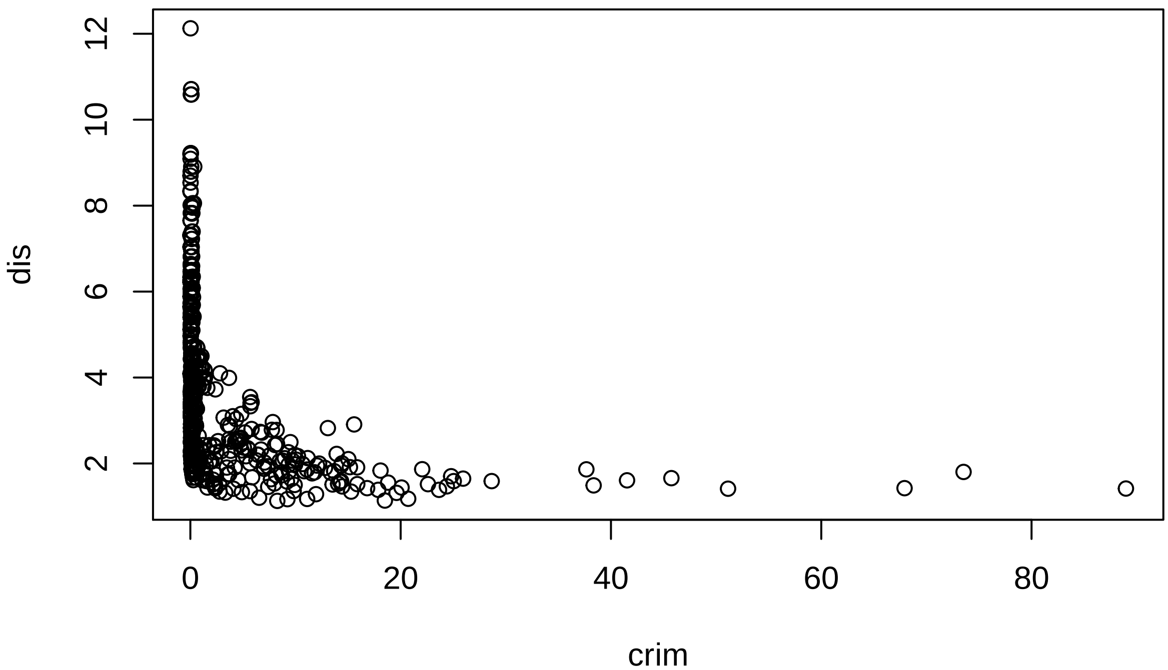
chas
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80
crim



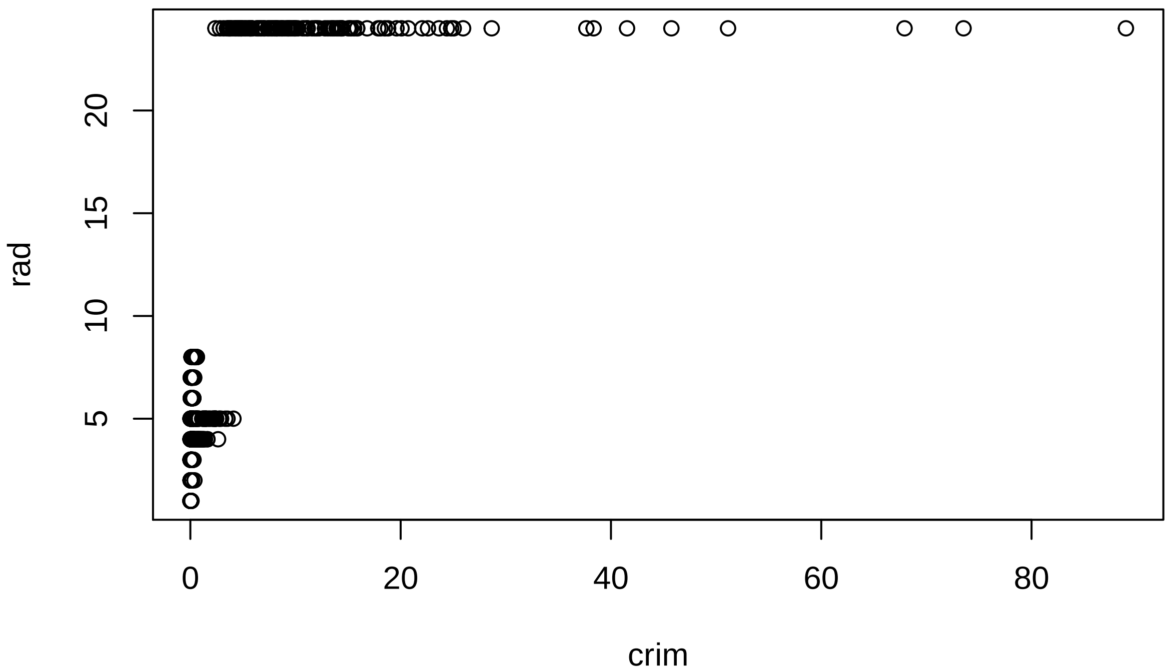
nox
0.4
0.5
0.6
0.7
0.8
0
20
40
60
80
crim

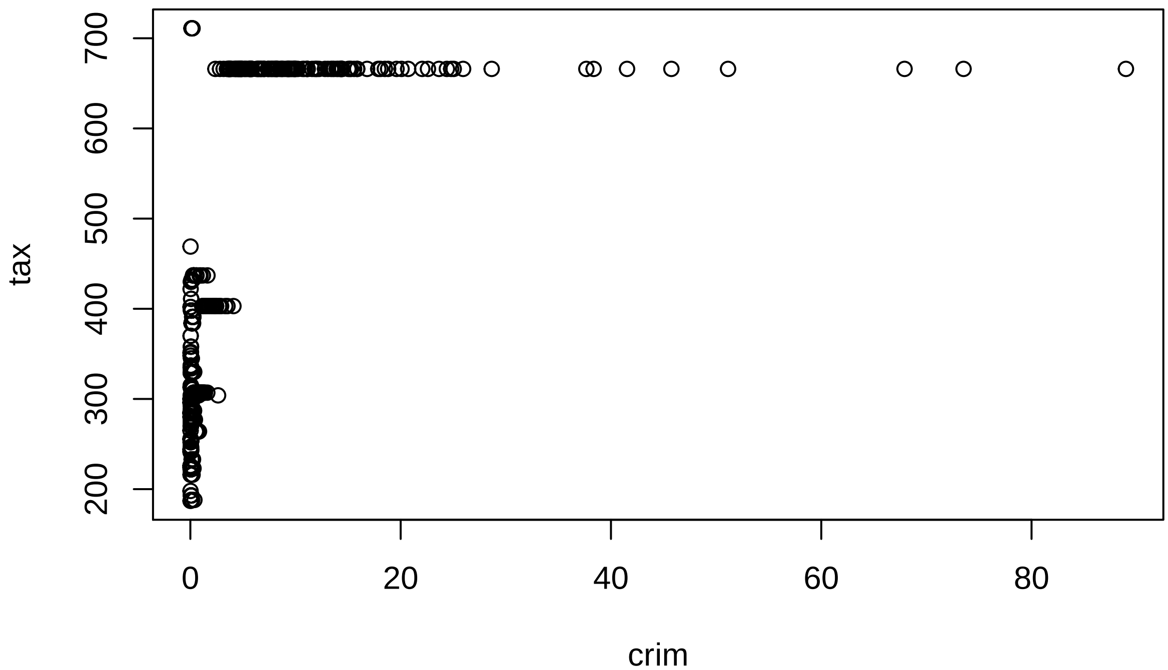

rm
8
7
6
5
4
0
20
40
60
80
crim

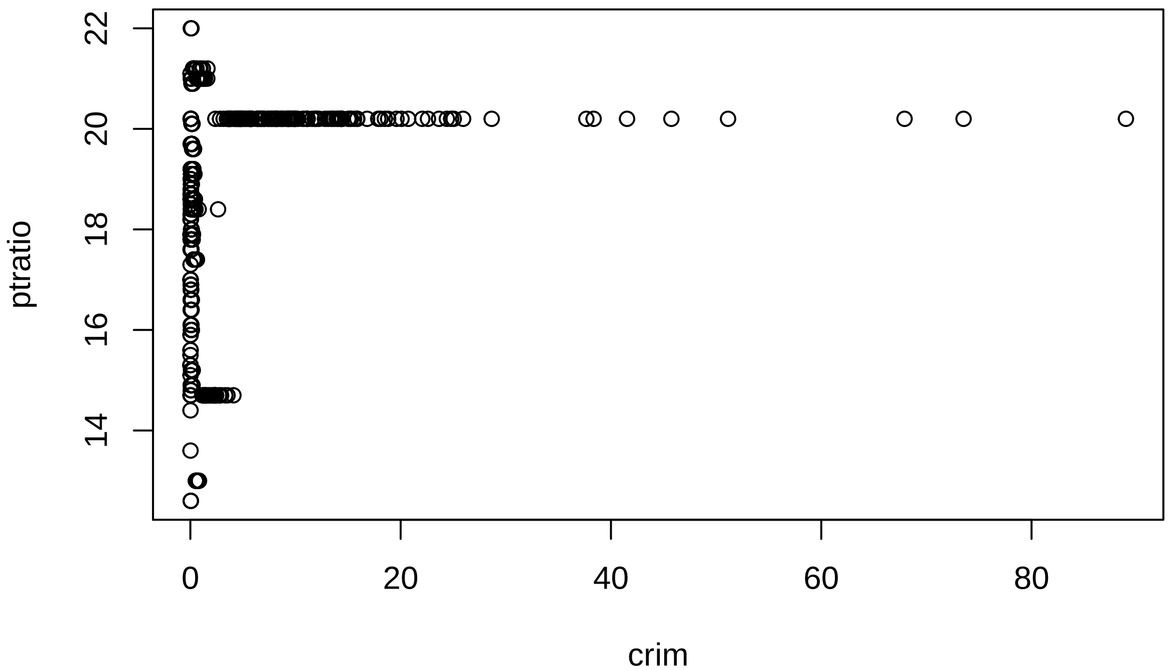

age
crim
0
20
40
60
80
100
0
20
40
60
80

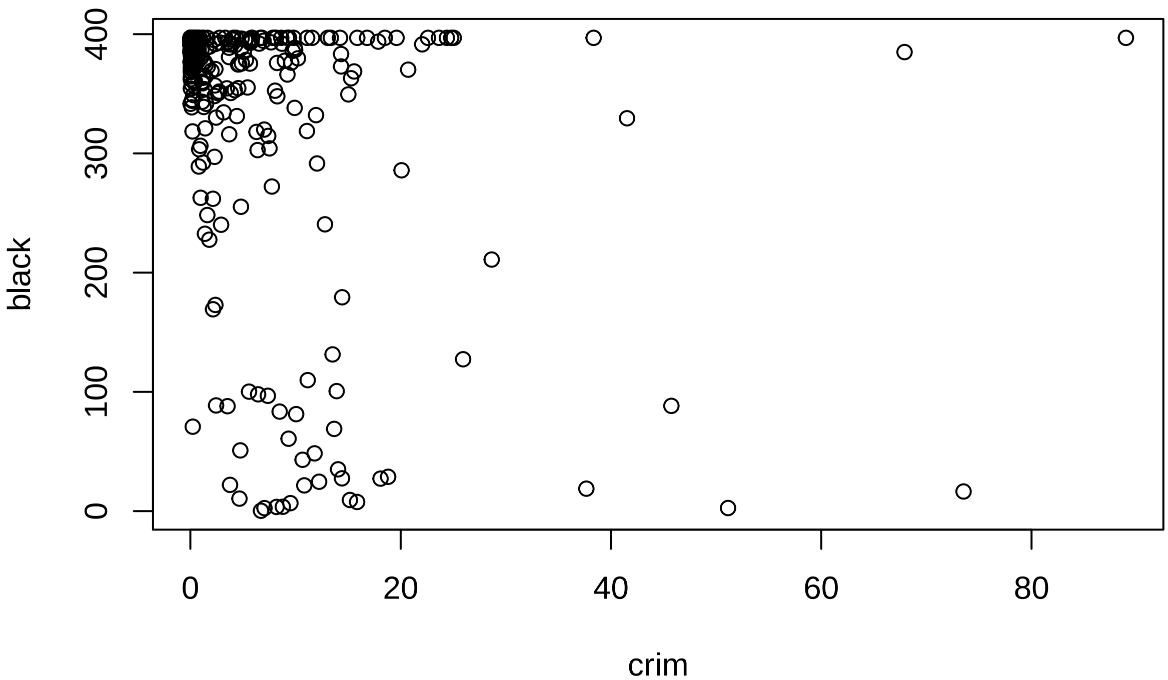


dis
crim

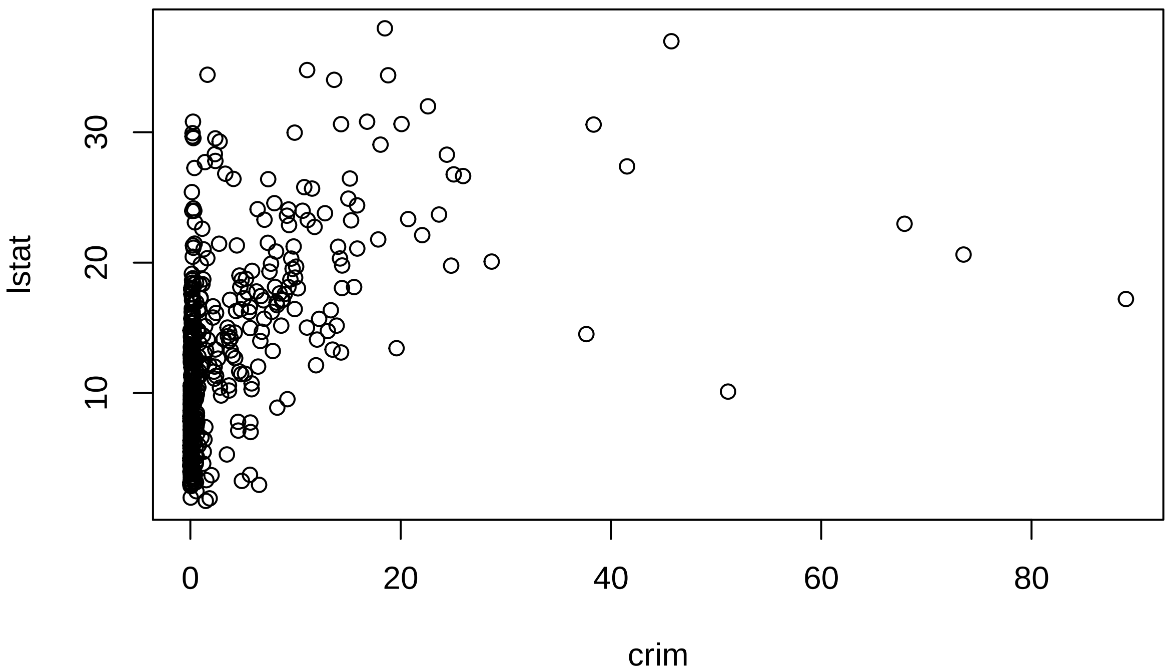

rad
20
15
10
5
0
20
40
60
80
crim

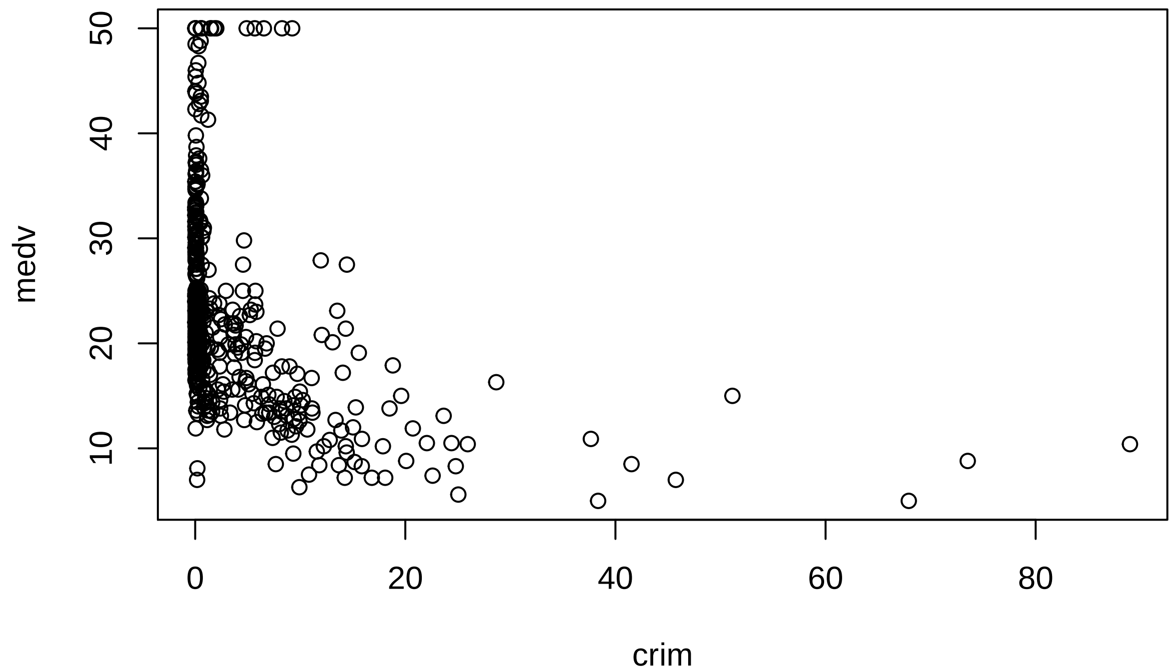

tax
200
300
400
500
600
700
0
20
40
60
80
crim


22
20
18
16
14
ptratio
0
20
40
60
80
crim



400
300
200
100
0
black
0
20
40
60
80
crim


lstat
30
20
10
0
20
40
60
80
crim



Among all the predictors, "zn", "indus", "nox", "rm", "age", "dis", "rad", "tax", "ptratio", "black", "lstat", "medv" are statistically significantly associated with per capita crime rate.

(b)

```
model.11 <- lm(crim ~ ., data = boston)
summary(model.11)
```

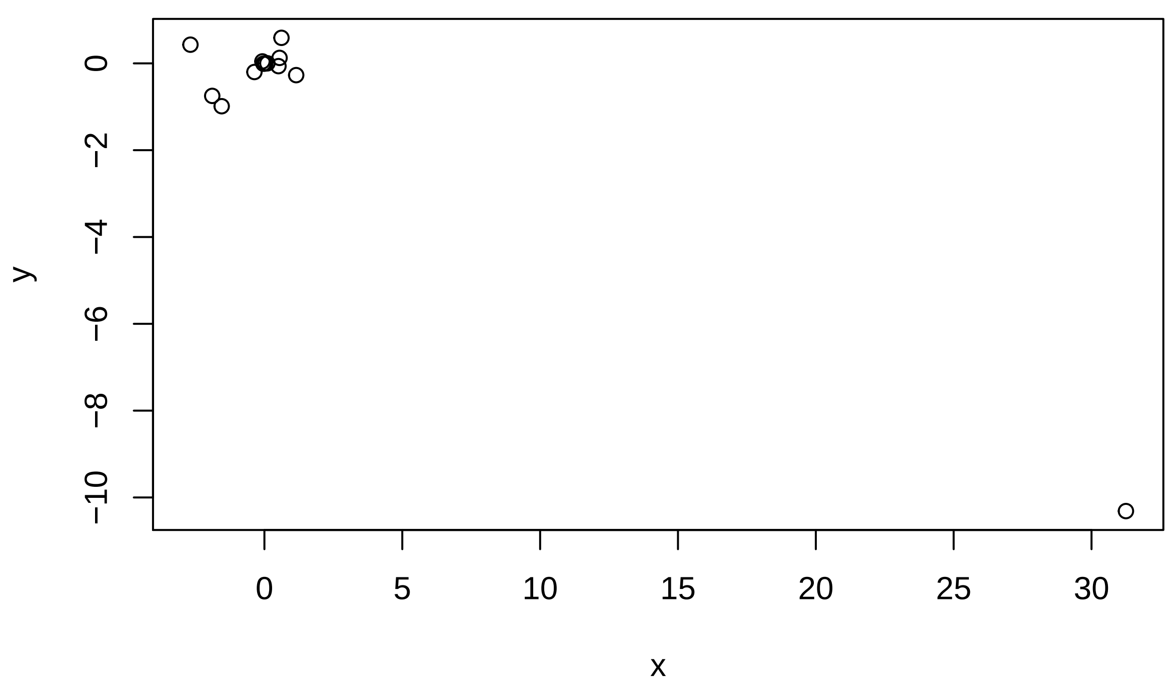
```
##
## Call:
## lm(formula = crim ~ ., data = boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.924 -2.120 -0.353  1.019 75.051
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.033228   7.234903   2.354 0.018949 *
## zn            0.044855   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
```

Since p-values of “zn”, “dis”, “rad”, “black” and “medv” are less than 0.05, we can reject the null hypothesis that coefficients of these parameters are 0.

(c)

```
y <- as.numeric(coef(model.11)[2:length(coef(model.11))])
plot(x,y)
```



Coefficient of “nox” is 31.249 in simple regression but -10.313535 in multiple regression.