36-401 Homework **2**

1 Problem 1

(a) Using law of total probability for the expectation, we have that

$$E[\epsilon] = \sum_{i=1}^{n} E[\epsilon|X_i] = \sum_{i=1}^{n} 0 = 0$$

Similarly, using the law of total variance covered in lecture note 1, we have that

$$Var[\epsilon] = E[Var(\epsilon|X)] + Var(E[Y|X]) = E[\sigma^2] + Var(0) = \sigma^2$$

(b) To find the least squares estimators, we want to find the best $\hat{\beta_1}$ and $\hat{\beta_0}$ to minimize $\sum_i (Y_i - [\beta_0 + \beta_1 X_i])^2$ To find this, we take the derivative and set the derivative to 0. Starting with $\hat{\beta_0}$,and using chain rule we have

$$\frac{d}{d\hat{\beta_0}} = \sum_i 2 * (Y_i - [\hat{\beta_0} + \hat{\beta_1} X_i]) * -1 = -2 \sum_i (Y_i - \hat{\beta_0} - \hat{\beta_1} X_i)$$

Setting this to 0 and simplifying we have that

$$\begin{aligned}
-2 \sum_i (Y_i - \hat{\beta_0} - \hat{\beta_1} X_i) = 0 \to \sum_i (Y_i - \hat{\beta_0} - \hat{\beta_1} X_i) = 0 &\to \sum_i Y_i - \sum_i \hat{\beta_0} - \sum_i \hat{\beta_1} X_i = 0 \\
&\to \sum_i (Y_i) - n\hat{\beta_0} - \hat{\beta_1} \sum_i X_i = 0 \to n\hat{\beta_0} = \sum_i Y_i - \hat{\beta_1} \sum_i X_i \\
&\to \hat{\beta_0} = \frac{1}{n} \sum_i Y_i - \frac{1}{n} \hat{\beta_1} \sum_i X_i = \bar{Y} - \hat{\beta_1} \bar{X}
\end{aligned}$$

So we have that $\hat{\beta} = \bar{Y} - \hat{\beta_1}\bar{X}$.
Similarly, we will take the derivative and set it to 0 to find the $\hat{\beta_1}$ that minimizes the sum. Using the chain rule we have

$$\frac{d}{d\hat{\beta_1}} = \sum_i 2 * (Y_i - [\hat{\beta_0} + \hat{\beta_1} X_i]) * -X_i$$

First, simplifying it by setting it to 0 we have

$$-2 \sum_i (Y_i X_i - \hat{\beta_0} X_i - \hat{\beta_1} X_i^2) = 0 \to \sum_i Y_i X_i - \hat{\beta_0} X_i - \hat{\beta_1} X_i^2 = 0$$

plugging in the minimum value for $\hat{\beta_0}$, $\hat{\beta_0}$ that we calcualted previously, we have that the left side is

$$\begin{aligned}
\sum_i Y_i X_i - \hat{\beta_0} X_i - \hat{\beta_1} X_i^2 = \sum_i Y_i X_i - (\bar{Y} - \hat{\beta_1}\bar{X}) X_i - \hat{\beta_1} X_i^2 = \sum_i Y_i X_i - \bar{Y} X_i + \hat{\beta_1} \bar{X} X_i - \hat{\beta_1} X_i^2 \\
= (\sum_i Y_i X_i - \bar{Y} X_i) - \hat{\beta_1} (\sum_i X_i^2 - \bar{X} X_i) = 0
\end{aligned}$$

So we have that

$$\hat{\beta}_1 = \frac{\sum_i Y_i X_i - \bar{Y} X_i}{\sum_i X_i^2 - \bar{X} X_i} = \frac{\sum_i X_i(Y_i - \bar{Y})}{\sum_i X_i(X_i - \bar{X})}$$

From, lecture notes, we know that $\sum_i Y_i - \bar{Y} = 0$. As $\bar{X}$ is a constant value, we can multiply this to the entire sum and let it go inside the summation. So we have that $\sum_i \bar{X}(Y_i - \bar{Y}) = 0$. For similar reasons, we have that $\sum_i \bar{X}(X_i - \bar{X}) = 0$.
Since each of these values, is 0 adding or subtracting these values will not change the result. So subtracting each value from the numerator and the denominator, we have that.

$$\hat{\beta}_1 = \frac{\sum_i X_i(Y_i - \bar{Y}) - \sum_i \bar{X}(Y_i - \bar{Y})}{\sum_i X_i(X_i - \bar{X}) - \sum_i \bar{X}(X_i - \bar{X})} = \frac{\sum_i X_i(Y_i - \bar{Y}) - \bar{X}(Y_i - \bar{Y})}{\sum_i X_i(X_i - \bar{X}) - \bar{X}(X_i - \bar{X})} = \frac{\sum_i (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_i (X_i - \bar{X})^2}$$

Thus we found the values for both $\hat{\beta}_0$ and $\hat{\beta}_1$

(c) First simplifying the expected value by using the definition of $\bar{Y}$ as given in the lecutre notes, we have that

$$E[\hat{\beta}_0 | X_1 ... X_n] = E[\bar{Y} - \hat{\beta}_1 \bar{X}] = E[\beta_0 + \beta_1 \bar{X} + \bar{\epsilon} - \hat{\beta}_1 \bar{X}]$$

Using linearity of expected values and our value for $E[\beta_1]$ that we calculated during class, we have that

$$E[\beta_0] + \bar{X} E[\beta_1] + E[\bar{\epsilon}] - \bar{X} E[\hat{\beta}_1] = \beta_0 + \bar{X}\beta_1 + 0 - \bar{X}\beta_1 = \beta_0$$

Then for the variance, we start with the same step.

$$Var[\hat{\beta}_0 | X_1 ... X_n] = Var[\bar{Y} - \hat{\beta}_1 \bar{X}] = Var[\beta_0 + \beta_1 \bar{X} + \bar{\epsilon} - \hat{\beta}_1 \bar{X}]$$

Similarly, since $\beta_0, \beta_1, \bar{X}$ are all constants, we have

$$Var[\bar{\epsilon}] + \bar{X}^2 Var[\hat{\beta}_1]$$

Finally, since we know the variance of β_1 from lecture and that $Var[\epsilon] = \sigma^2$ from above, we can plug it in this to get

$$\frac{\sigma^2}{n} + \bar{X}^2 (\frac{\sigma^2}{n s_X^2}) = \frac{\sigma^2 (s_X^2 + \bar{X}^2)}{n s_X^2}$$

(d) Plugging in the definition we got in part b for $\hat{\beta}_0$ we have that

$$\sum_i \hat{\epsilon}_i = \sum_i Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i = \sum_i Y_i - (\bar{Y} - \hat{\beta}_1 \bar{X}) - \hat{\beta}_1 X_i = (\sum_i Y_i - \bar{Y}) - \hat{\beta}_1 (\sum_i \bar{X} - X_i) = 0 - 0 = 0$$

2 Problem 2

(a) Similarly to the first question, we try to minimize the squared difference between the real and the regression value, which is $\sum_i (Y_i - [\beta_1 X_i])^2$. This can be done again by taking the derivative and setting it to 0 to find the minimum, so

$$\frac{d}{d\hat{\beta}_1} = \sum_i 2*(Y_i - \hat{\beta}_1 X_i) * -X_i = -2\sum_i (Y_i X_i - \hat{\beta}_1 X_i^2)$$

Setting this equal to 0 and simplifying we have

$$-2\sum_i Y_i X_i - \hat{\beta}_1 X_i^2 = 0 \rightarrow \sum_i Y_i X_i - \hat{\beta}_1 X_i^2 = 0 \rightarrow \sum_i Y_i X_i - \hat{\beta}_1 \sum_i X_i^2 = 0$$

Finally we have that

$$\hat{\beta}_1 = \frac{\sum_i Y_i X_i}{\sum_i X_i^2}$$

(b) To start, we will first try to simplify $\hat{\beta}_1$

$$\begin{aligned}\hat{\beta}_1 &= \frac{\sum_i Y_i X_i}{\sum_i X_i^2} = \frac{\sum_i X_i(\beta_1 X_i + \epsilon_i)}{\sum_i X_i^2} = \frac{\sum_i \beta_1 X_i^2 + X_i \epsilon_i}{\sum_i X_i^2} = \frac{\sum_i \beta_1 X_i^2 + \sum_i X_i \epsilon_i}{\sum_i X_i^2} \\ &= \frac{{}_1 \sum_i X_i^2 + \sum_i X_i \epsilon_i}{\sum_i X_i^2} = \beta_1 + \frac{\sum_i X_i \epsilon_i}{\sum_i X_i^2}\end{aligned}$$

When taking the expected value of this, we can use that each of X_i are constants because we are conditioning on their values, linearity of expected values, and the fact that $E[\epsilon_i] = 0$ to get

$$E[\hat{\beta}_1] = E[\beta_1 + \frac{\sum_i X_i \epsilon_i}{\sum_i X_i^2}] = E[\beta_1] + \frac{1}{\sum_i X_i^2}\sum_i X_i E[\epsilon_i] = \beta_1 + \frac{1}{\sum_i X_i^2}\sum_i X_i * 0 = \beta_1$$

When getting the variance, we will work on the simplified $\hat{\beta}_1$ value to get

$$\begin{aligned}Var[\hat{\beta}_1] &= Var[\beta_1 + \frac{\sum_i X_i \epsilon_i}{\sum_i X_i^2}] = Var[\beta_1] + Var[\frac{\sum_i X_i \epsilon_i}{\sum_i X_i^2}] = \frac{\sum_i X_i^2 Var[\epsilon_i]}{(\sum_i X_i^2)^2} \\ &= \frac{\sum_i X_i^2 \sigma^2}{(\sum_i X_i^2)^2} = \frac{\sigma^2 \sum_i X_i^2}{(\sum_i X_i^2)^2} = \frac{\sigma^2}{\sum_i X_i^2}\end{aligned}$$

3 Problem 5

(a) Multiplying and simplifying we get

$$\begin{aligned} H^2 = H * H &= (X(X^TX)^{-1}X^T)(X(X^TX)^{-1}X^T) \\ &= X(X^TX)^{-1}(X^TX)(X^TX)^{-1}X^T \\ &= X(X^TX)^{-1}X^T = H \end{aligned}$$

So since $H^2 = H$, H must idempotent.

(b) According to the property of traces we have that for some matrices A, B, C, $trace(ABC) = trace(CAB)$. Setting $A = X, B = (X^TX)^{-1}, C = X^T$ we have that

$$trace(ABC) = trace(CAB) \rightarrow trace(X(X^TX)^{-1}X^T) = trace(X^TX(X^TX)^{-1} \rightarrow trace(H) = trace(I)$$

Since the identity matrix has ones on its main diagonal and its size is same as (X^TX), we have that its trace must be the number of columns, which is one more than the number of covariates, so we have that $trace(H) = d + 1$

(c) We will start by finding the middle value, $(X^TX)^{-1}$ We know that X^T is

$$\begin{bmatrix} 1 & 1 & ... & 1 \\ X_1 & X_2 & ... & X_n \end{bmatrix}$$

multiplying that with x we get a $2X2$ matrix

$$\begin{bmatrix} n & \sum_i X_i \\ \sum_i X_i & \sum_1 X_i^2 \end{bmatrix}$$

To invert this matrix, we want to find the determinant which is

$$ad - bc = n * \sum_1 X_i^2 - \sum_i X_i * \sum_i X_i = n \sum_1 X_i^2 - (\sum_1 X_i)^2$$

As the result of this is just a scalar value, we will take it out of the matrix multiplication and add it back in later on. So we have that the inverse of this matrix without the determinant division is

$$\begin{bmatrix} \sum_1 X_i^2 & -\sum_i X_i \\ -\sum_i X_i & n \end{bmatrix}$$

multiplying X with this we get

$$\begin{bmatrix} \sum_i X_i^2 - X_1 \sum_i X_i & -\sum_i X_i + nX_1 \\ \sum_i X_i^2 - X_2 \sum_i X_i & -\sum_i X_i - nX_2 \\ \sum_i X_i^2 - X_3 \sum_i X_i & -\sum_i X_i - nX_3 \\ ... & ... \\ \sum_i X_i^2 - X_n \sum_i X_i & -\sum_i X_i - nX_n \end{bmatrix}$$

Finally we will multiply this with X^T to get

$$\begin{bmatrix} \sum_i X_i^2 - 2 * X_1 \sum_i X_i + nX_1^2 & \sum_i X_i^2 - X_1 \sum_i X_i - X_2 \sum_i X_i + nX_1X_2 & .. & \sum_i X_i^2 - X_1 \sum_i X_i - X_n \sum_i X_i + nX_1X_n \\ \sum_i X_i^2 - X_2 \sum_i X_i - X_1 \sum_i X_i + nX_1X_2 & \sum_i X_i^2 - 2 * X_2 \sum_i X_i + nX_2^2 & .. & \sum_i X_i^2 - X_2 \sum_i X_i - X_n \sum_i X_i + nX_2X_n \\ .. & .. & .. & .. \\ \sum_i X_i^2 - X_n \sum_i X_i - X_1 \sum_i X_i + nX_1X_n & \sum_i X_i^2 - X_n \sum_i X_i - X_2 \sum_i X_i + nX_2X_n & .. & \sum_i X_i^2 - 2 * X_n \sum_i X_i + nX_n^2 \end{bmatrix}$$

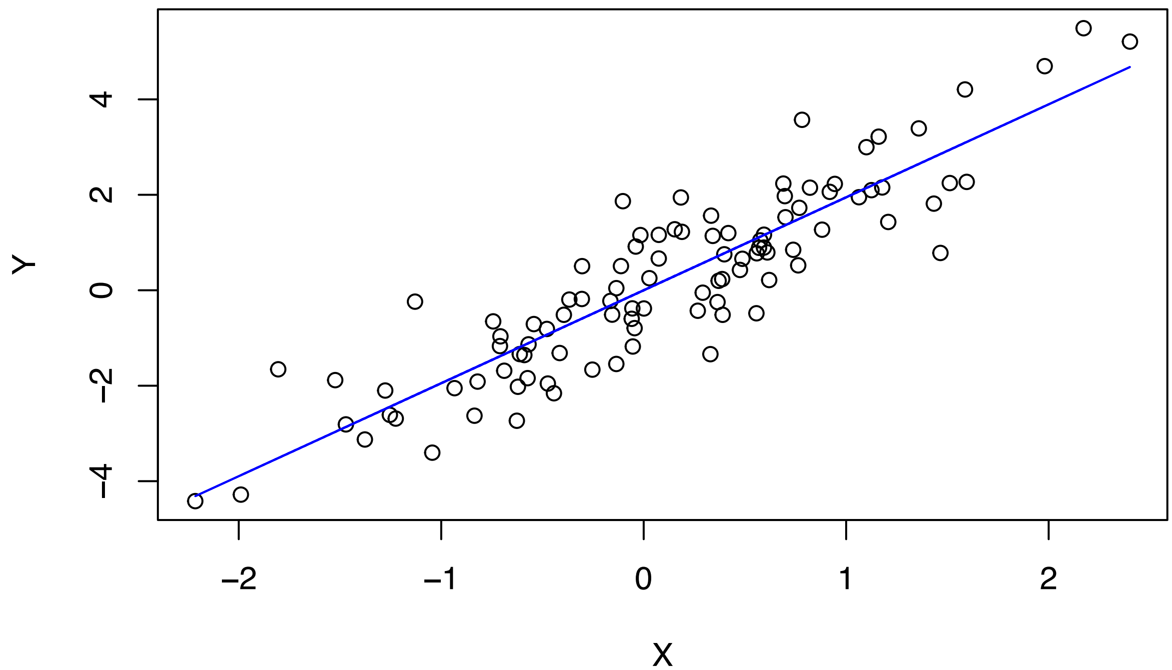
And the result would be this matrix divided by d. And because there is no simplification, the resulting element would just be each element in the above matrix divided by d. So the result is

$$\begin{bmatrix} \frac{\sum_i X_i^2-2*X_1\sum_i X_i+nX_1^2}{n\sum_1 X_i^2-(\sum_1 X_i)^2} & \frac{\sum_i X_i^2-X_1\sum_i X_i-X_2\sum_i X_i+nX_1X_2}{n\sum_1 X_i^2-(\sum_1 X_i)^2} & .. & \frac{\sum_i X_i^2-X_1\sum_i X_i-X_n\sum_i X_i+nX_1X_n}{n\sum_1 X_i^2-(\sum_1 X_i)^2} \\ \frac{\sum_i X_i^2-X_2\sum_i X_i-X_1\sum_i X_i+nX_1X_2}{n\sum_1 X_i^2-(\sum_1 X_i)^2} & \frac{\sum_i X_i^2-2*X_2\sum_i X_i+nX_2^2}{n\sum_1 X_i^2-(\sum_1 X_i)^2} & .. & \frac{\sum_i X_i^2-X_2\sum_i X_i-X_n\sum_i X_i+nX_2X_n}{n\sum_1 X_i^2-(\sum_1 X_i)^2} \\ .. & .. & .. & .. \\ \frac{\sum_i X_i^2-X_n\sum_i X_i-X_1\sum_i X_i+nX_1X_n}{n\sum_1 X_i^2-(\sum_1 X_i)^2} & \frac{\sum_i X_i^2-X_n\sum_i X_i-X_2\sum_i X_i+nX_2X_n}{n\sum_1 X_i^2-(\sum_1 X_i)^2} & .. & \frac{\sum_i X_i^2-2*X_n\sum_i X_i+nX_n^2}{n\sum_1 X_i^2-(\sum_1 X_i)^2} \end{bmatrix}$$

Homework **2** - 36401

2(c)

We can force R to exclude an intercept by adding -1 to the end of the x value. For my random numbers, the estimated slope given by this regression line going through the origin is 1.94767 and its standard error is 0.09601.




```
##
## Call:
## lm(formula = Y ~ X - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0726 -0.5393 -0.1125  0.5235  2.0667
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## X  1.94767    0.09601   20.29   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8644 on 99 degrees of freedom
## Multiple R-squared:  0.8061, Adjusted R-squared:  0.8041
## F-statistic: 411.5 on 1 and 99 DF,  p-value: < 2.2e-16
```

Now, if we run the regression with an intercept this time, we get the slope of 1.95114 and a standard error of 0.09717.

```r
out_intercept = lm(Y ~ X)
summary(out_intercept)
```

```
##
## Call:
## lm(formula = Y ~ X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.05180 -0.51505 -0.08527  0.54780  2.09286
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.02585    0.08748  -0.295    0.768
## X            1.95114    0.09717  20.080   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8684 on 98 degrees of freedom
## Multiple R-squared:  0.8045, Adjusted R-squared:  0.8025
## F-statistic: 403.2 on 1 and 98 DF,  p-value: < 2.2e-16
```

3(a)

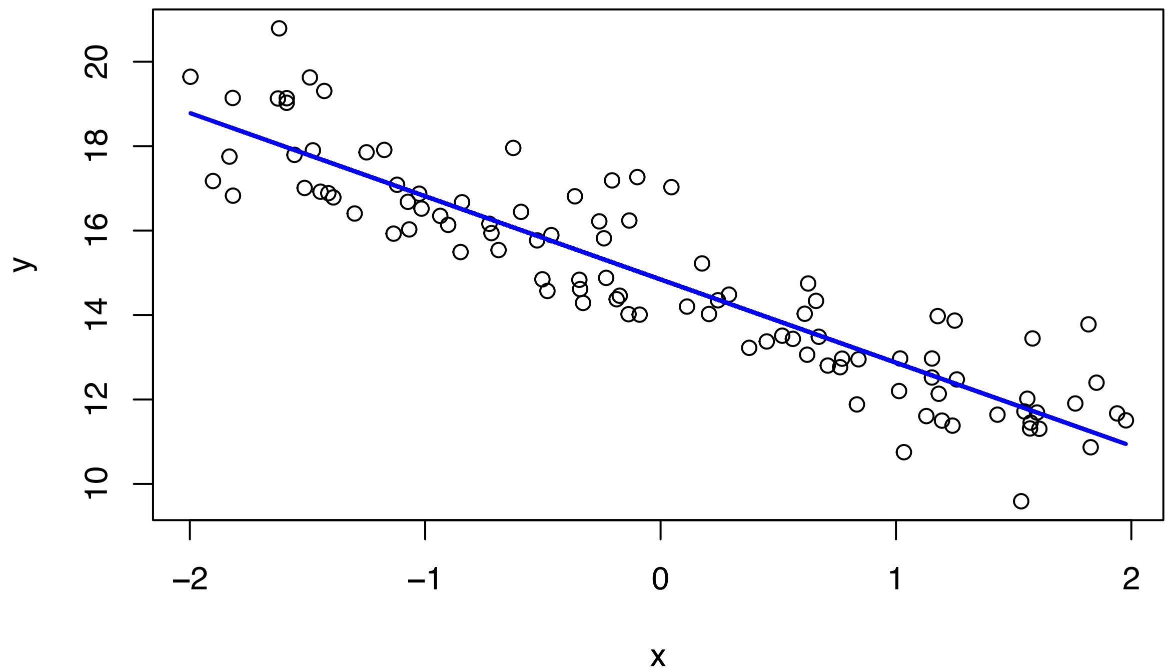
We plotted the data, residuals, standardized residuals, and the cooks distance. With our random data, our linear regression gave a slope and intercept estimate of -1.97101 and 14.84341 which is very close to the true value of -2 and 15. Looking at just the data and the regression line, the line actually seems to fit the data very well with a seemingly balanced number of points above and below the line and most of the points following the shape of the line. Looking at the residual plot, we can see that there is no pattern with seemingly a balanced number of points above and below the 0 line. However, there are some points that seem very high. Taking a look at the studentized residual, it does not seem too different from the normal residuals so we can see the same trend. There is no pattern in the points and we can see that the high points we observed were almost 3 standard deviations away. Looking at the cooks distance, the highest distance is still less than 0.12 which is good as no particular data point has an immense effect on the data, but we can see that the high points that were far away from 0 in the residual plots have the highest influence. Finally, the q-q plot, we can see that the values are very linear showing that the distribution of the residuals is very close to a normal distribution.

```r
n = 100
set.seed(123)
x = runif(n,-2,2)
set.seed(1234)
```

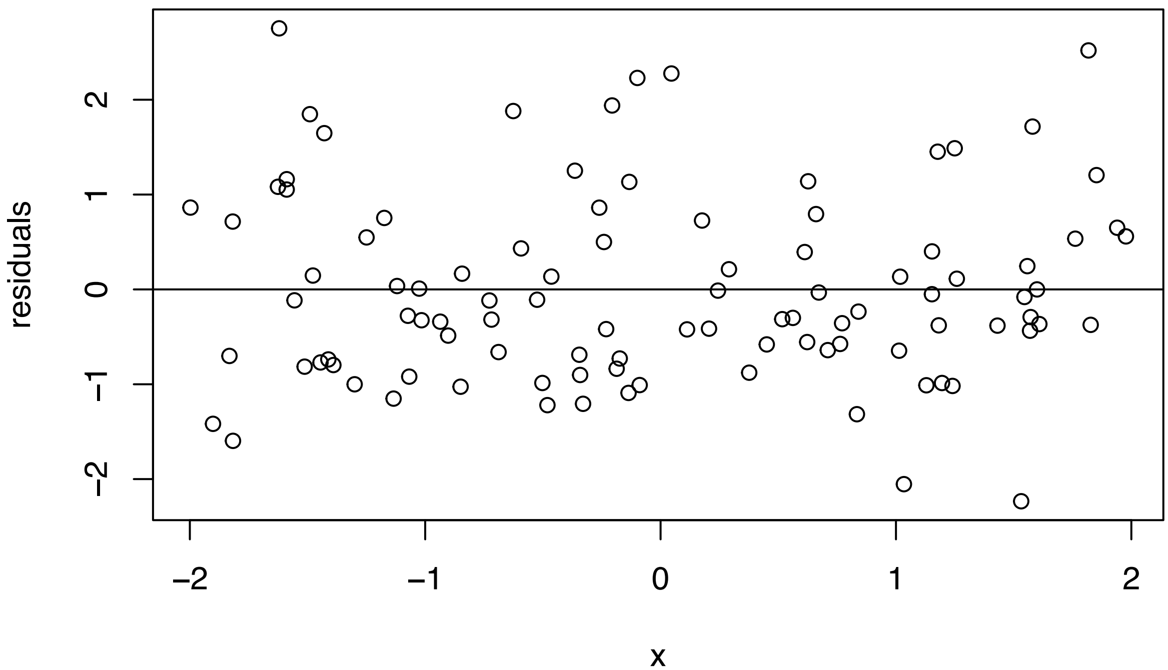
```r
eps = rnorm(n,0,1)
y = 15 - 2*x + eps

plot(y~x)
out = lm(y ~ x)
summary(out)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2335 -0.7082 -0.2562  0.5817  2.7526
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 14.84341    0.10090  147.11   <2e-16 ***
## x           -1.97101    0.08895  -22.16   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.009 on 98 degrees of freedom
## Multiple R-squared:  0.8336, Adjusted R-squared:  0.8319
## F-statistic:   491 on 1 and 98 DF,  p-value: < 2.2e-16
```

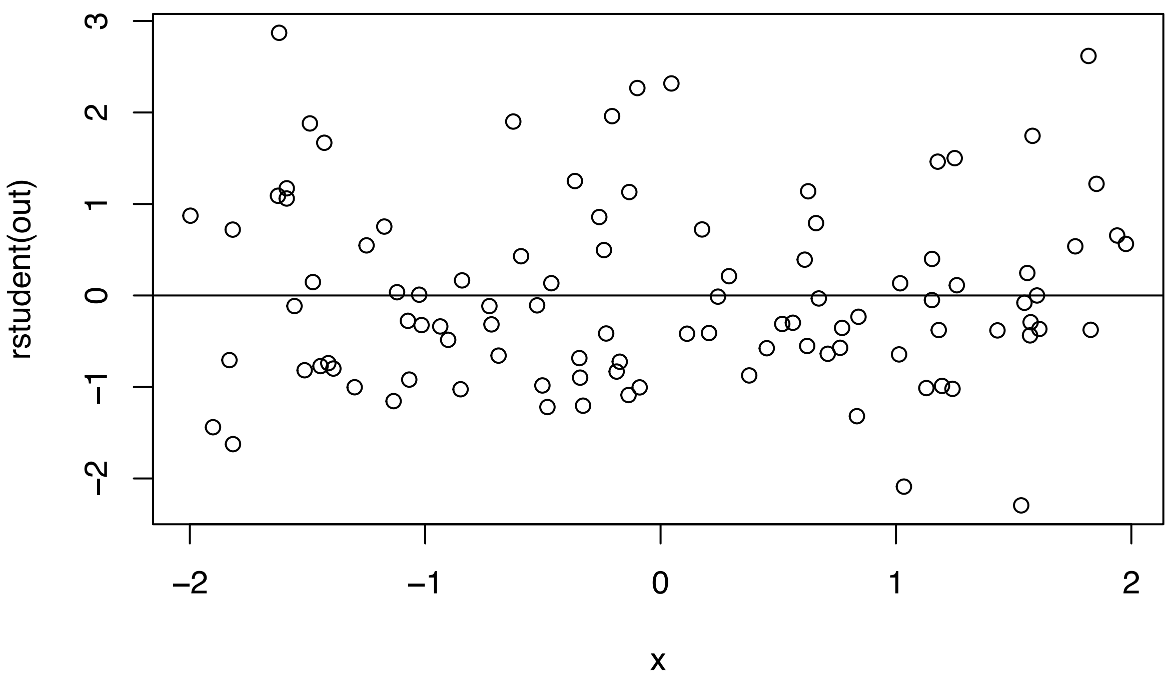
```r
lines(x,fitted(out),lwd=2,col="blue")
```



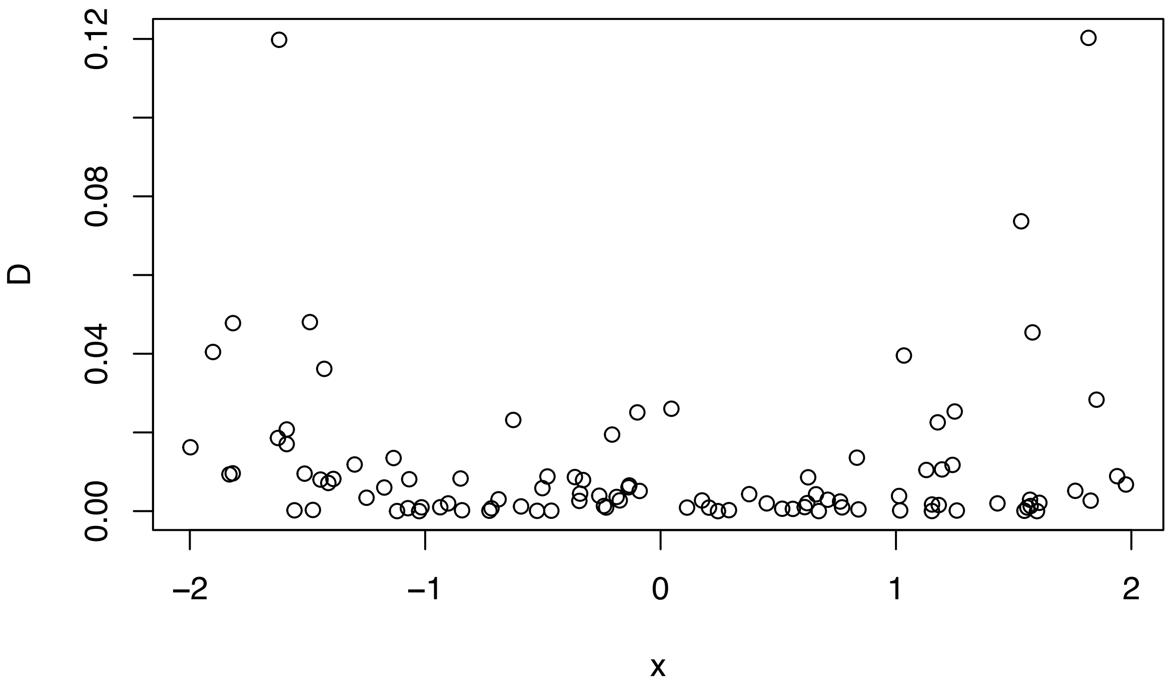

```
residuals = y - fitted(out) #residual plot
plot(x,residuals)
abline(h=0)
```



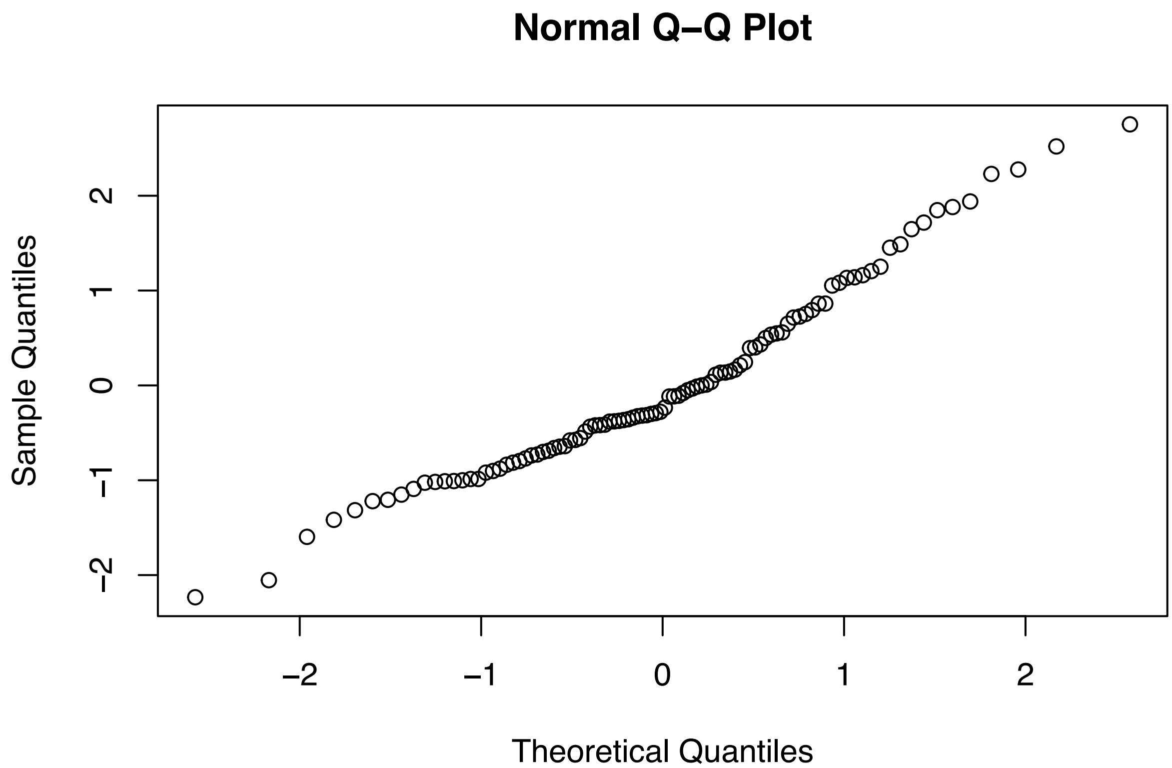

```r
plot(x,rstudent(out)) #standardized residuals
abline(h=0)
```




```
D = cooks.distance(out)
plot(x,D) #cooks distance
```




```
qqnorm(residuals) # normal q-q
```



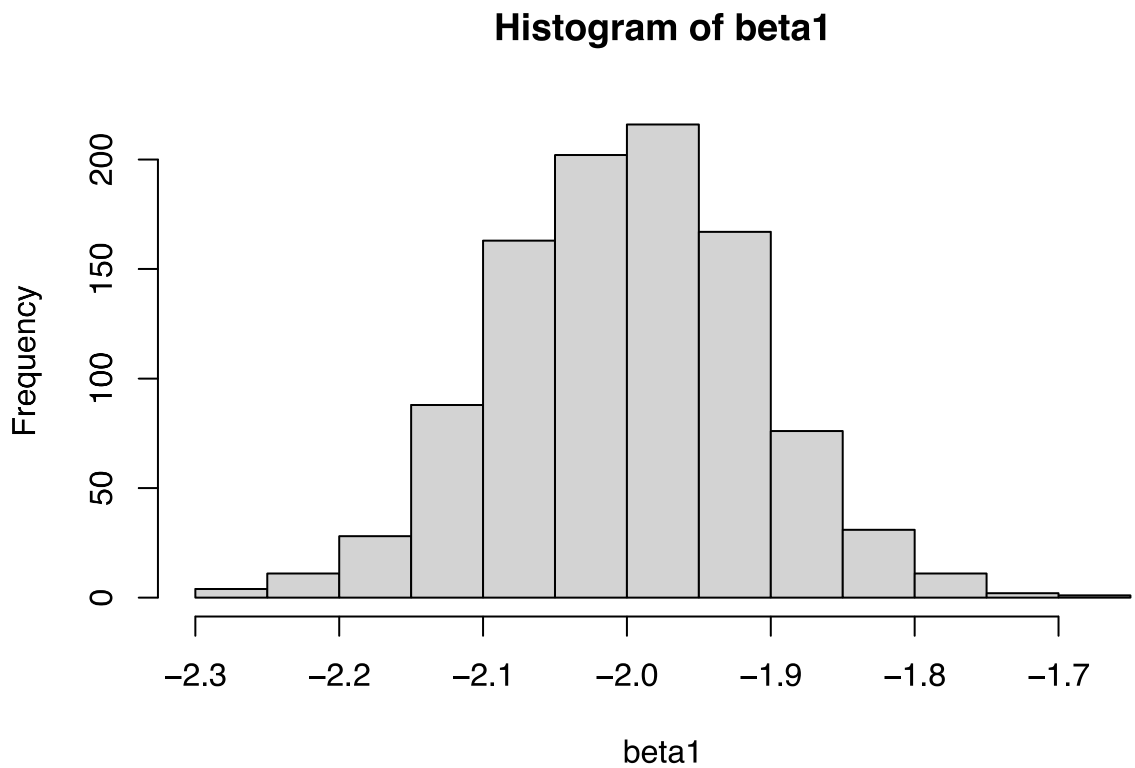
3(b)

We got the mean of -2.000801 from the beta1s. We expected it to be around -2 which is the true value for beta1 because the value that r uses for lm is from the formula in number 1 of homework, which we got in class to be an unbiased estimator for beta1. So we expect the mean or the expected value of all the beta hats produced by r to be around beta1.

```
nsim = 1000
beta1 = rep(0,1000)
n = 100
for(i in 1:nsim){
  set.seed(i)
  x = runif(n,-2,2)
  set.seed(i+1)
  eps = rnorm(n,0,1)
  y = 15 - 2*x + eps
  beta1[i] = lm(y~x)$coef[2]
}
mean(beta1)
```

```
## [1] -2.000801
```

```
hist(beta1)
```



4(a)

```
#setup
library(wooldridge)
data(hprice2)
attach(hprice2)
str(hprice2)
```

```
## 'data.frame':    506 obs. of  12 variables:
##  $ price   : num  24000 21599 34700 33400 36199 ...
##  $ crime   : num  0.006 0.027 0.027 0.032 0.069 ...
##  $ nox     : num  5.38 4.69 4.69 4.58 4.58 ...
##  $ rooms   : num  6.57 6.42 7.18 7 7.15 ...
##  $ dist    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ radial  : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ proptax : num  29.6 24.2 24.2 22.2 22.2 ...
##  $ stratio : num  15.3 17.8 17.8 18.7 18.7 ...
##  $ lowstat : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ lprice  : num  10.09 9.98 10.45 10.42 10.5 ...
##  $ lnox    : num  1.68 1.55 1.55 1.52 1.52 ...
##  $ lproptax: num  5.69 5.49 5.49 5.4 5.4 ...
##  - attr(*, "time.stamp")= chr "25 Jun 2011 23:03"
```

```
names(hprice2)
```

```
##  [1] "price"    "crime"    "nox"      "rooms"    "dist"     "radial"
```

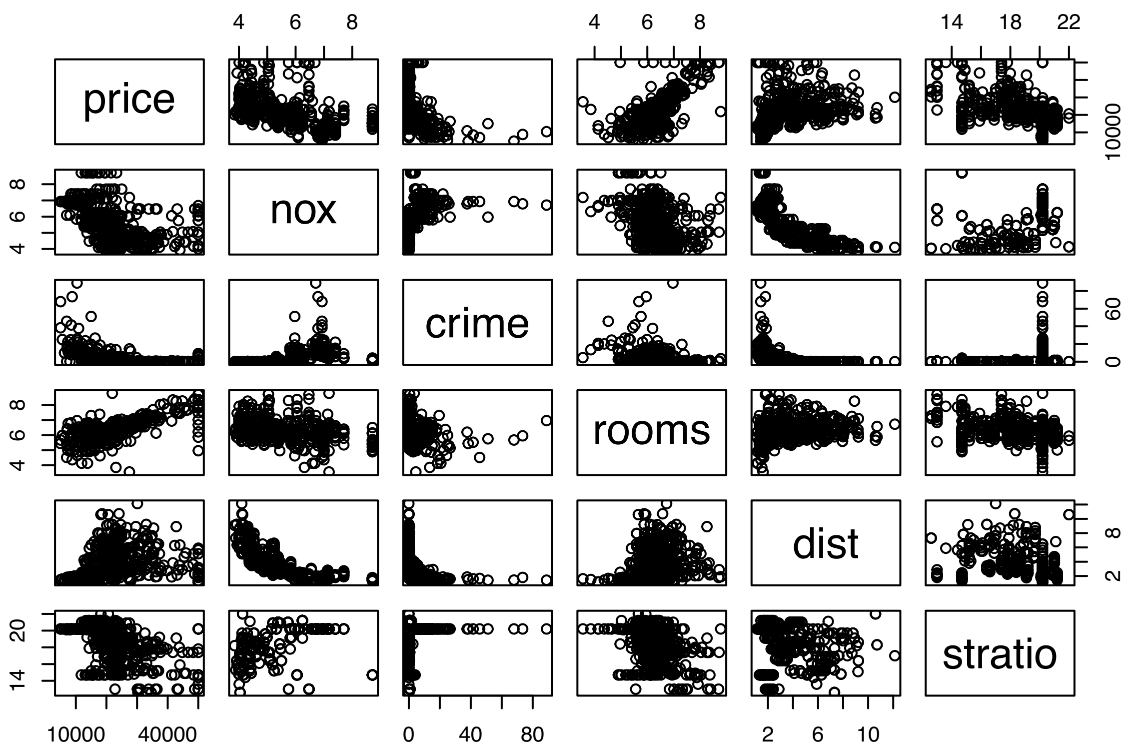
```
##  [7] "proptax"  "stratio"  "lowstat"  "lprice"   "lnox"     "lproptax"
```

```
help(hprice2)
```

```
## starting httpd help server ... done
```

Price and rooms seem to have a pretty distinct relation with median price of housing increasing as the average number of rooms increase. Also,nitrous oxide concentration seems to have a relation with distance with nox decreasing as distance to employment centers increase. While there is not much data in the higher levels, and the data is all clumped together there also seems to be a general relationship between crime and price with crime going down as the median housing price goes up.

```
df = data.frame(price,nox,crime,rooms,dist,stratio)
pairs(df)
```



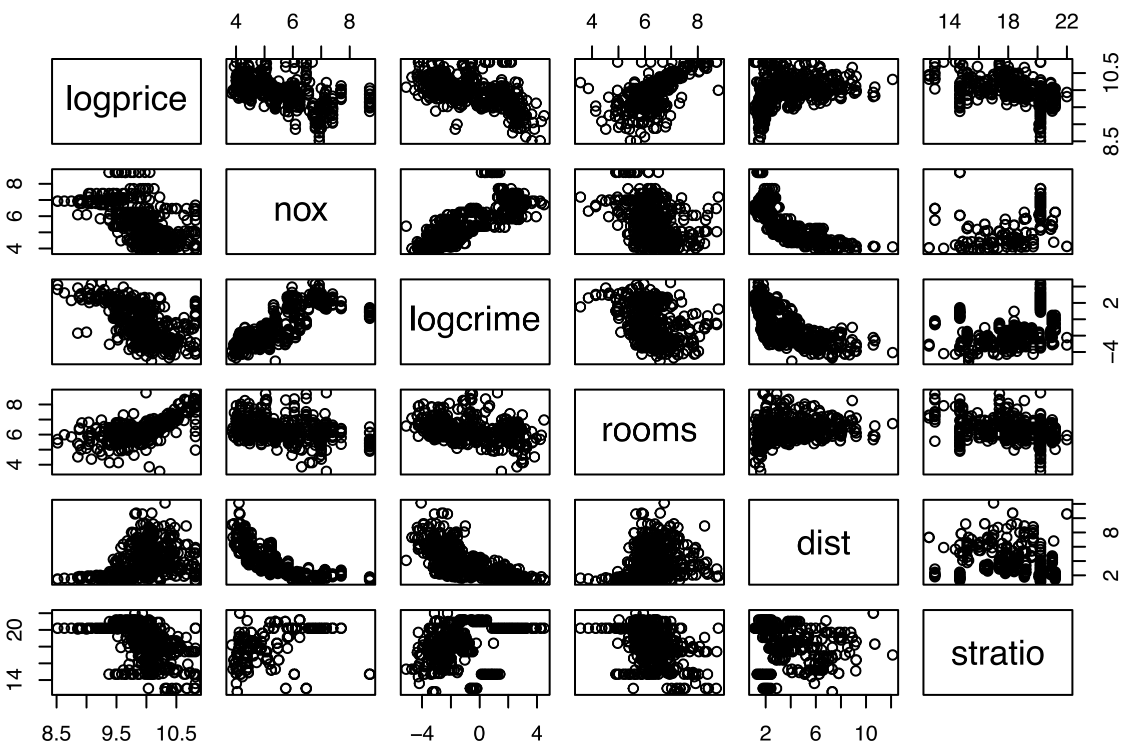

```
summary(df)
```

```
##      price            nox            crime              rooms      
##  Min.   : 5000   Min.   :3.85   Min.   : 0.0060   Min.   :3.560  
##  1st Qu.:16850   1st Qu.:4.49   1st Qu.: 0.0820   1st Qu.:5.883  
##  Median :21200   Median :5.38   Median : 0.2565   Median :6.210  
##  Mean   :22512   Mean   :5.55   Mean   : 3.6115   Mean   :6.284  
##  3rd Qu.:24999   3rd Qu.:6.24   3rd Qu.: 3.6770   3rd Qu.:6.620  
##  Max.   :50001   Max.   :8.71   Max.   :88.9760   Max.   :8.780  
```

```
##       dist          stratio
##  Min.   : 1.130   Min.   :12.60
##  1st Qu.: 2.100   1st Qu.:17.40
##  Median : 3.210   Median :19.10
##  Mean   : 3.796   Mean   :18.46
##  3rd Qu.: 5.188   3rd Qu.:20.20
##  Max.   :12.130   Max.   :22.00
```

4(b)

```
logcrime = log(crime)
logprice = log(price)
df = data.frame(logprice, nox, logcrime, rooms, dist, stratio)
pairs(df)
```



4(c) Estimated values for beta0 hat to beta5 hat is (9.913692, -0.102621, -0.045677, 0.246691, -0.036770, -0.046014). sigma^2 or residual standard error is 0.2573.

```
out = lm(logprice ~ nox + logcrime + rooms + dist + stratio)
summary(out)
```

```
##
## Call:
## lm(formula = logprice ~ nox + logcrime + rooms + dist + stratio)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.91463 -0.12544  0.00265  0.12188  1.37855
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.913692   0.254190  39.001  < 2e-16 ***
## nox         -0.102621   0.019567  -5.245 2.32e-07 ***
## logcrime    -0.045677   0.009481  -4.818 1.93e-06 ***
## rooms        0.246691   0.018113  13.620  < 2e-16 ***
## dist        -0.036770   0.008742  -4.206 3.08e-05 ***
## stratio     -0.046014   0.006164  -7.465 3.73e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2573 on 500 degrees of freedom
## Multiple R-squared:  0.6088, Adjusted R-squared:  0.6048
## F-statistic: 155.6 on 5 and 500 DF,  p-value: < 2.2e-16
```

4(d)

None of the residual plots look truly random, the residual plot with fitted value and dist converges to 0 as it increases, residual plot of rooms have a dot in the middle, and even the other graphs do not look completely random. The maximum residual point has studentized value 5.549505 and happens at 369 according to which max function.

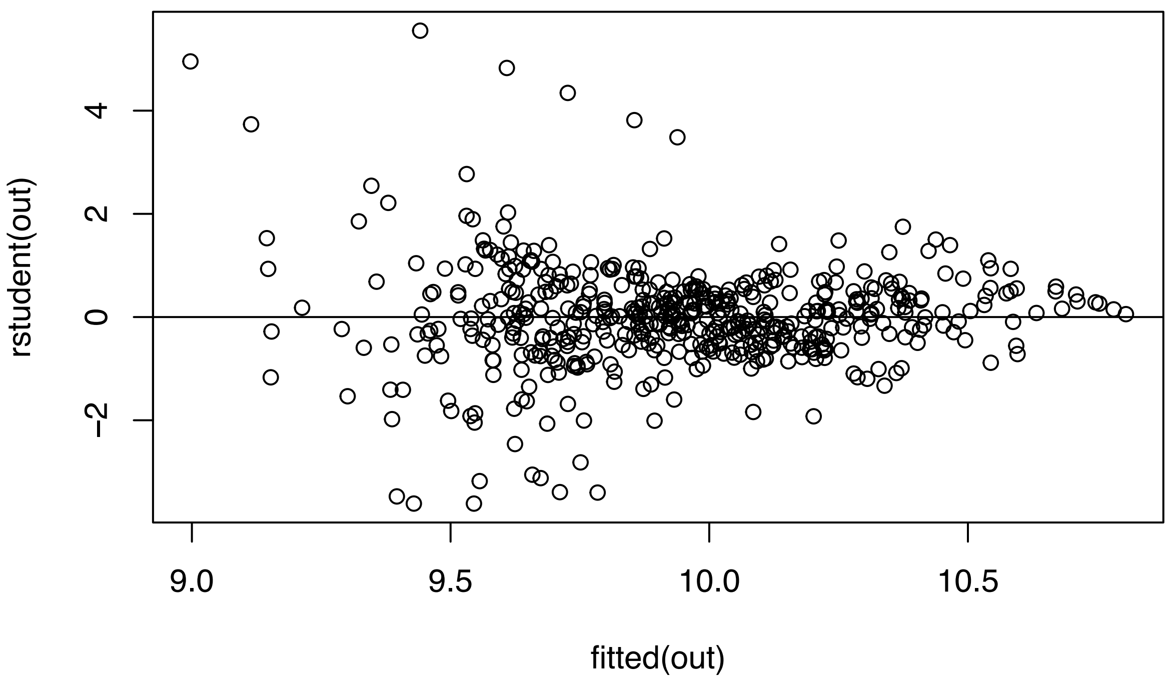
```r
out = lm(logprice ~ nox + logcrime + rooms + dist + stratio)
summary(out)
```

```
##
## Call:
## lm(formula = logprice ~ nox + logcrime + rooms + dist + stratio)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.91463 -0.12544  0.00265  0.12188  1.37855
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.913692   0.254190  39.001  < 2e-16 ***
## nox         -0.102621   0.019567  -5.245 2.32e-07 ***
## logcrime    -0.045677   0.009481  -4.818 1.93e-06 ***
## rooms        0.246691   0.018113  13.620  < 2e-16 ***
## dist        -0.036770   0.008742  -4.206 3.08e-05 ***
## stratio     -0.046014   0.006164  -7.465 3.73e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2573 on 500 degrees of freedom
## Multiple R-squared:  0.6088, Adjusted R-squared:  0.6048
## F-statistic: 155.6 on 5 and 500 DF,  p-value: < 2.2e-16
```

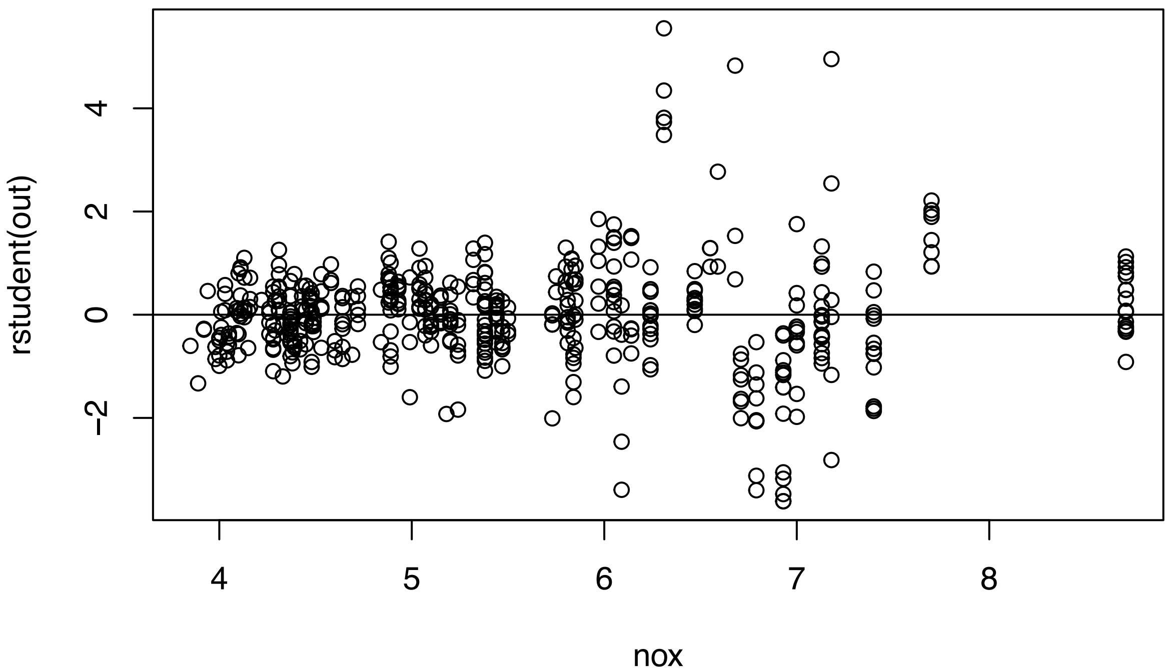
```r
which.max(rstudent(out))
```

```
## 369
## 369
```

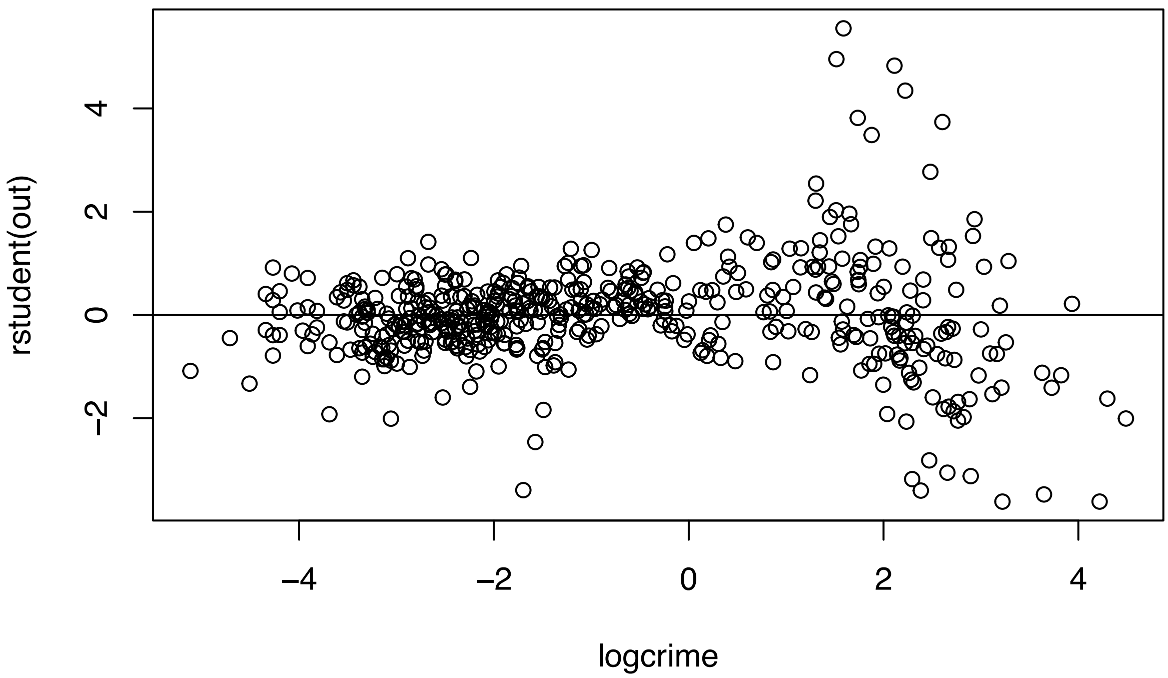
```r
plot(fitted(out),rstudent(out))
abline(h=0)
```



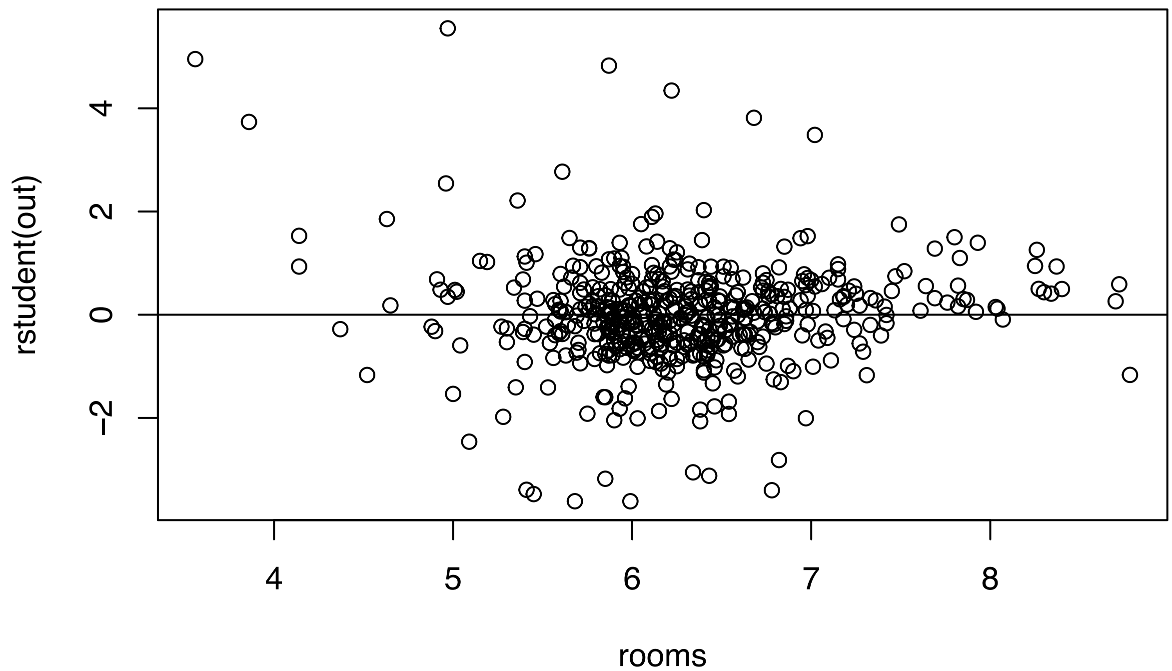

```r
plot(nox, rstudent(out))
abline(h=0)
```



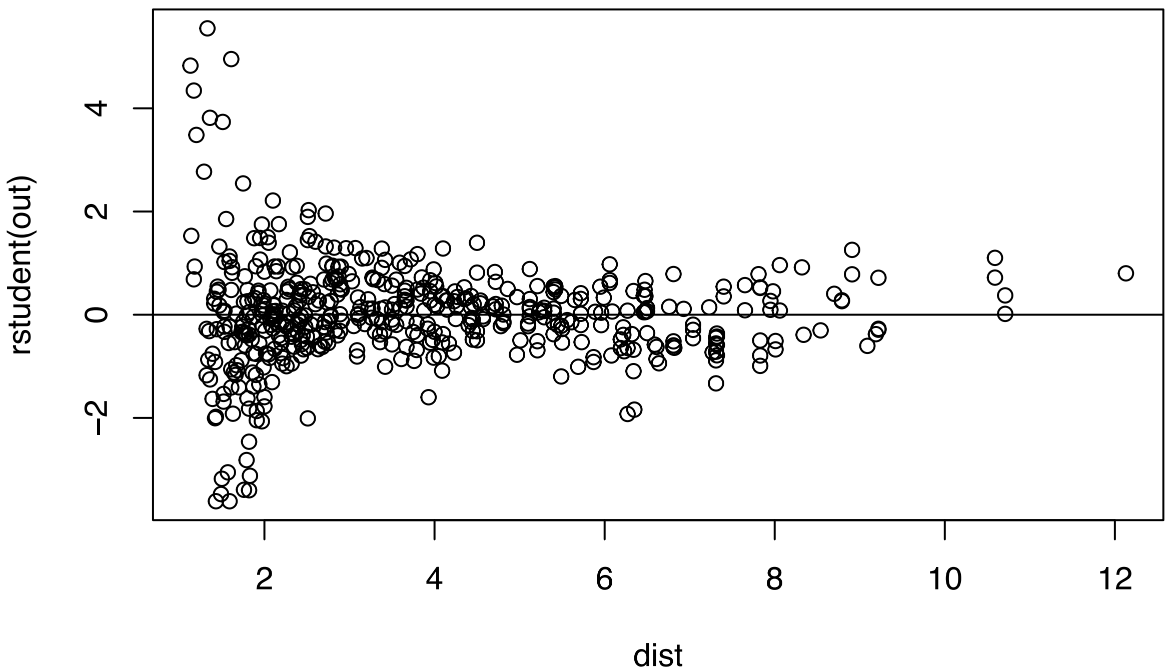

```
plot(logcrime, rstudent(out))
abline(h=0)
```



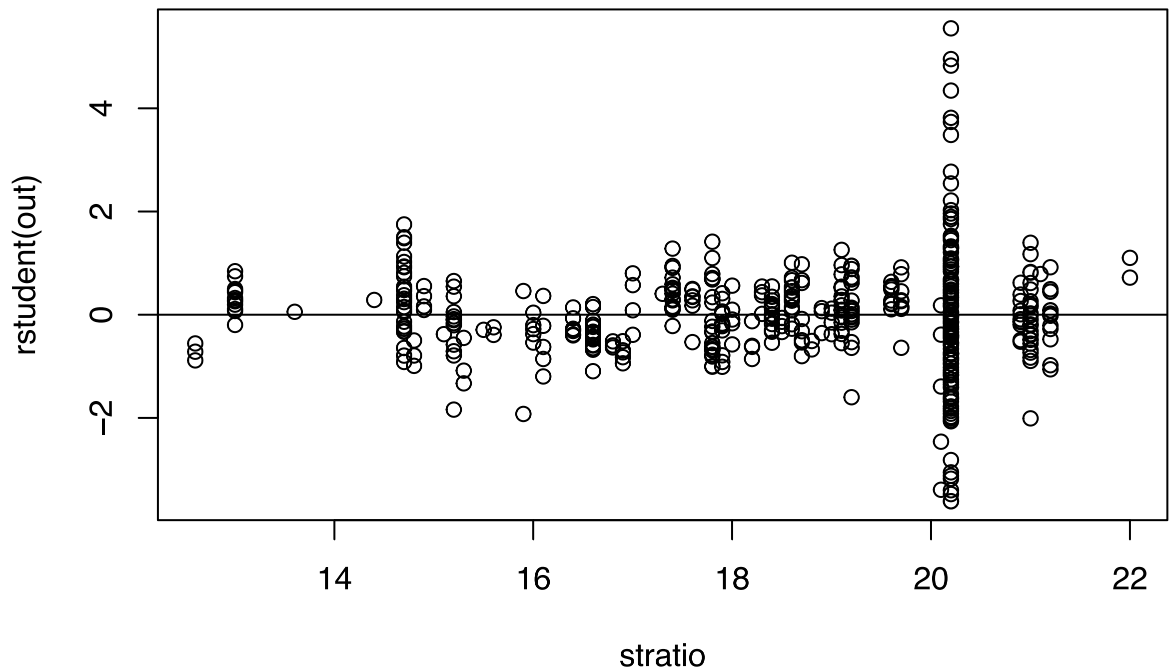

```
plot(rooms, rstudent(out))
abline(h=0)
```




```
plot(dist, rstudent(out))
abline(h=0)
```



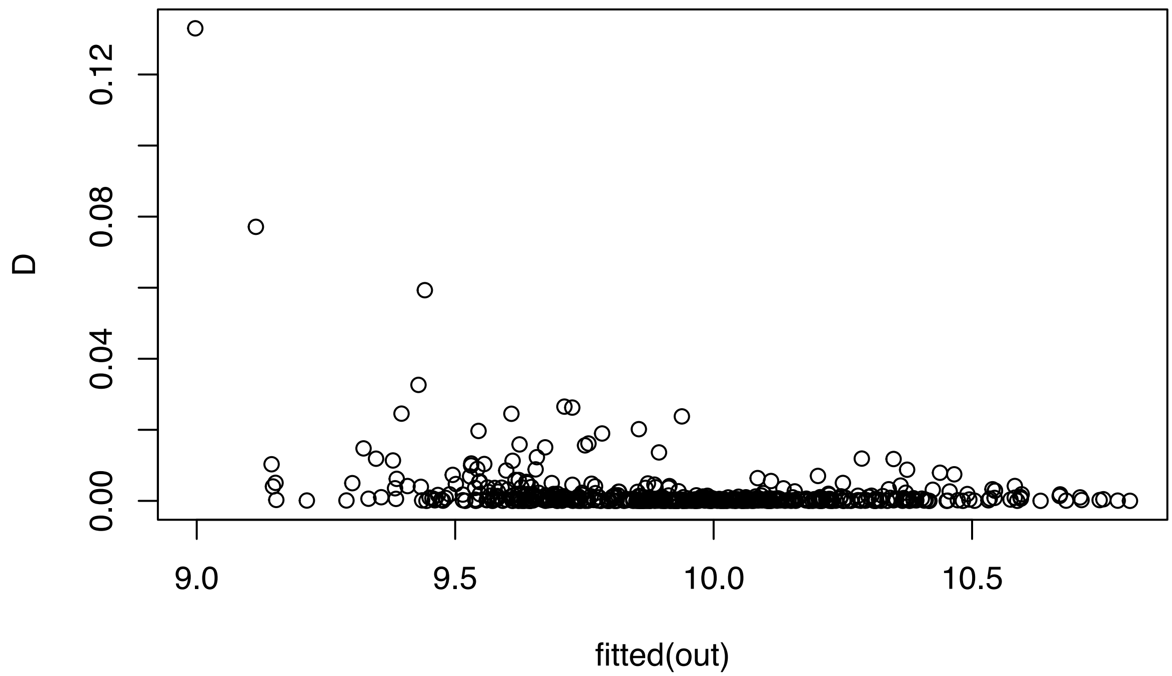

```
plot(stratio, rstudent(out))
abline(h=0)
```



4(e)

The 366 point has the largest influence according to the which.max function.

```
D = cooks.distance(out)
plot(fitted(out),D) #cooks distance
```




```
which.max(D)
```

```
## 366
## 366
```

4(f)

The 90% confidence intervals using the sandwich library is printed below.

```
library(sandwich)
V = vcovHC(out)
se = sqrt(diag(V))
alpha = 0.1
z = -qnorm(alpha/2)
left = out$coef -z*se
right = out$coef + z*se
print(cbind(left,right))
```

```
##                    left       right
## (Intercept)  9.43599872 10.39138551
## nox         -0.13427267 -0.07096873
## logcrime    -0.06352027 -0.02783305
## rooms        0.20417270  0.28920994
## dist        -0.04894755 -0.02459176
## stratio     -0.05370904 -0.03831919
```

4(g)

The prediction interval is printed below

```
newx = data.frame(nox=8,logcrime = 4, rooms = 6, dist = 3, stratio = 20)
predict(out, newdata = newx, interval = "prediction", level = 0.9)
```

```
##        fit      lwr      upr
## 1 9.359577 8.930912 9.788241
```