Manual_Solution_Exercise3

Gota Morishita

9/16/2021

Cenceptual

1

The null hypothese corresponding to TABLE 3.4 are

- TV advertisement has no association with sales.
- radio advertisement has no association with sales.
- newspaper advertisement has no association with sales.

Looking at TABLE 3.4, TV and radio has low p-values, so TV and radio have nonnegligible association with sales. On the other hand, newspaper does not association with sales.

2

KNN classifier is used for classification problem as you can guess from its name. The way to do that is to gather K points closest to a point you want to estimate and assign the point to the most common class among the K nearest points.

On the other hand, KNN regressoin is used for regression problem. The estimation procedure is similar to KNN classifier. First, gather K points closest to a point you want to estimate and assign the estimated point to averaged values of K observed response variables.

3

The linear model is as follows:

$$salary = 50 + 20 \times GPA + 0.07 \times IQ + 35 \times Gender + 0.01 \times GPA \times IQ - 10 \times GPA \times Gender$$

(a)

The correct answer is (iii).

With IQ and GPA fixed, $salary = (35 - 10 \times GPA) \times Gender + const.$. When GPA is high enough, the coefficient of Gender is negative, so males earn more money than female since the coding of Gender is 1 for female and 0 for male.

(b)

Substituting 1, 110 and 4 for Gender, IQ, and GPA, we have $salary = 50 + 80 + 7.7 + 35 + 4.4 - 40 = 137.1$

(c)

False. Small value of a coefficient does not mean little evidence of an effect while small p-value does.

4

(a)

We expect the cubic regression to have lower training RSS because there is a noise when you observe the data and the cubic regression is more complex, thus fitting the training data better than the linear regression.

(b)

We expect the linear regression to have lower test RSS. The cubic regression tends to fit the observed data too well to generalize.

(c)

The linear regression is a submodel of the cubic regression. Therefore, the training RSS of the cubic regression is always smaller than that of the linear regression.

(d)

There is not enough information to tell which model has lower training RSS. It depends on the true model generating the data.

5

$$a_{i'} = \frac{x_{i'} x_i}{\sum_k x_k^2}$$

6

From (3.4), we have $\bar{y} = \hat{\beta}_0 + \hat{\beta}_1 \bar{x}$, which completes the proof.

7

Assume that $\bar{y} = \bar{x} = 0$.

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$
$$TSS = \sum_{i=1}^{n} y_i^2$$

Our aim is to show that $R^2 = Cor(X, Y)^2$

$$\begin{aligned} R^2 &= 1 - RSS/TSS \\ &= 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i)^2} \\ \hat{y}_i &= \hat{\beta}_1 x_i \\ &= \frac{\sum x_i y_i}{\sum x_i^2} \end{aligned}$$

Substituting $\hat{y}_i$, we have

$$R^2 = \frac{\sum x_i y_i}{\sum x_i^2 \sum y_i^2}$$

Applied

Set up

```
library(ISLR)
library(ggplot2)
```

8

(a)

```
lm.fit <- lm(mpg ~ horsepower, data = Auto)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5710  -3.2592  -0.3435   2.7630  16.9240
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.935861   0.717499   55.66   <2e-16 ***
## horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.906 on 390 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

(i)

The p-value for horsepower is very low, so we can say that there is a (negative) relationship between the predictor and the response.

(ii)

The R^2 statistic is 0.6059, so the relationship is moderately strong.

(iii)

Negative.

(iv)

```
predict(lm.fit, data.frame(horsepower = c(98)), interval = 'confidence')
```

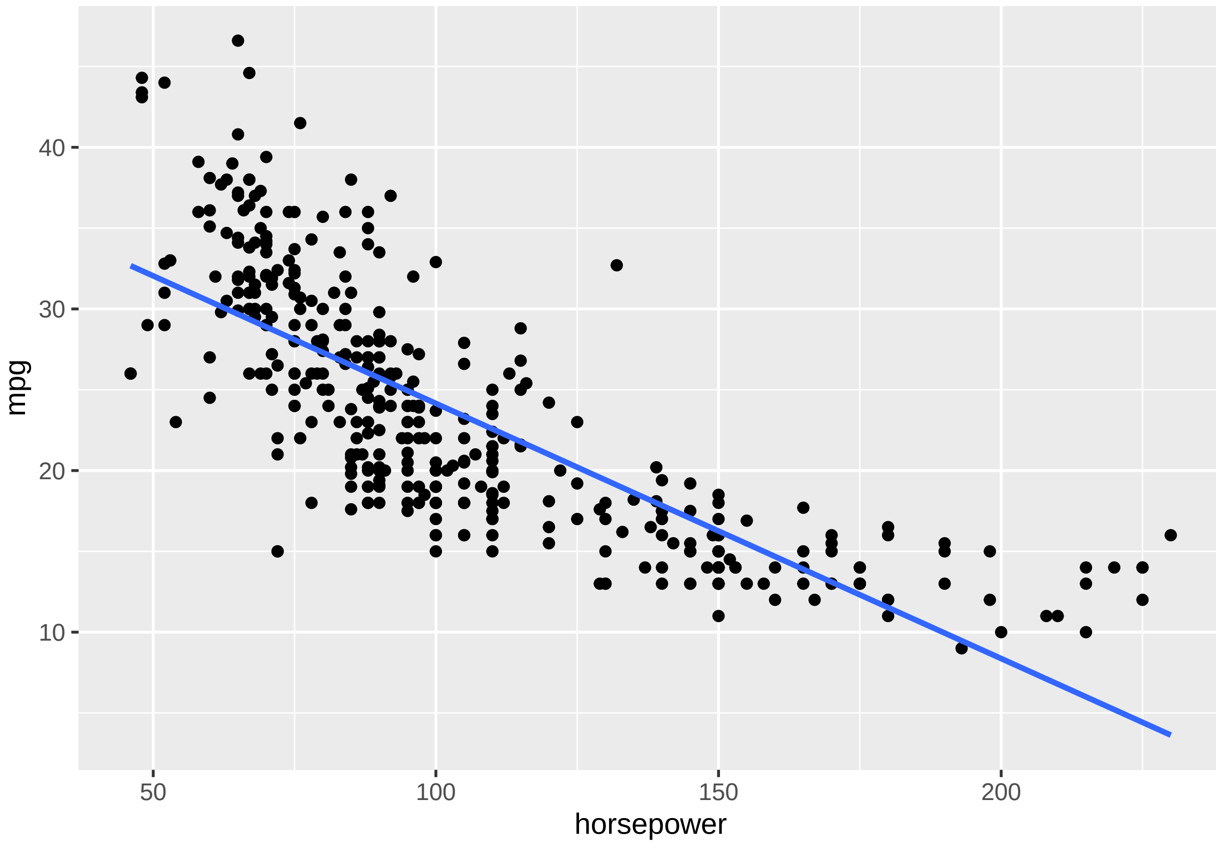
```
##        fit      lwr      upr
## 1 24.46708 23.97308 24.96108
```

```
predict(lm.fit, data.frame(horsepower = c(98)), interval = 'prediction')
```

```
##        fit     lwr      upr
## 1 24.46708 14.8094 34.12476
```

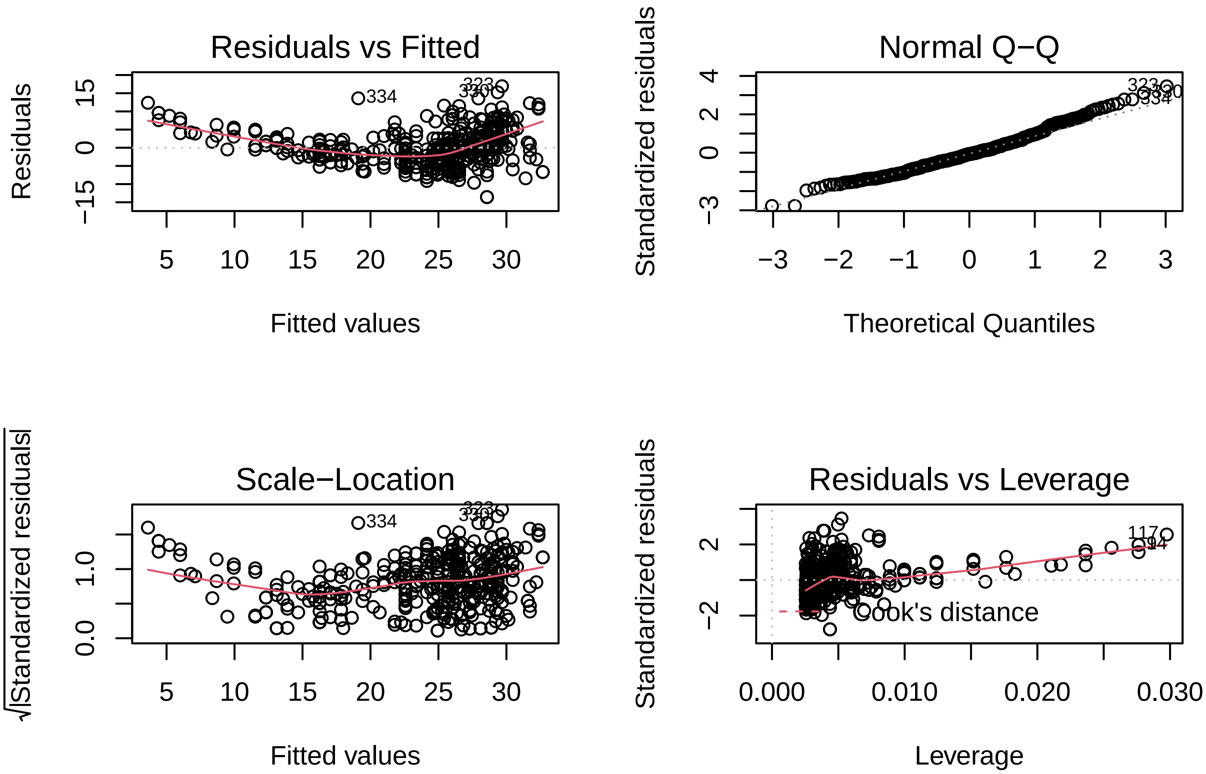
(b)

```
ggplot(data = Auto) +
  geom_point(mapping = aes(x = horsepower, y = mpg)) +
  geom_smooth(mapping = aes(x = horsepower, y = mpg), method = "lm", formula = y ~ x, se=FALSE)
```



(c)

```
par(mfrow=c(2,2))
plot(lm.fit)
```

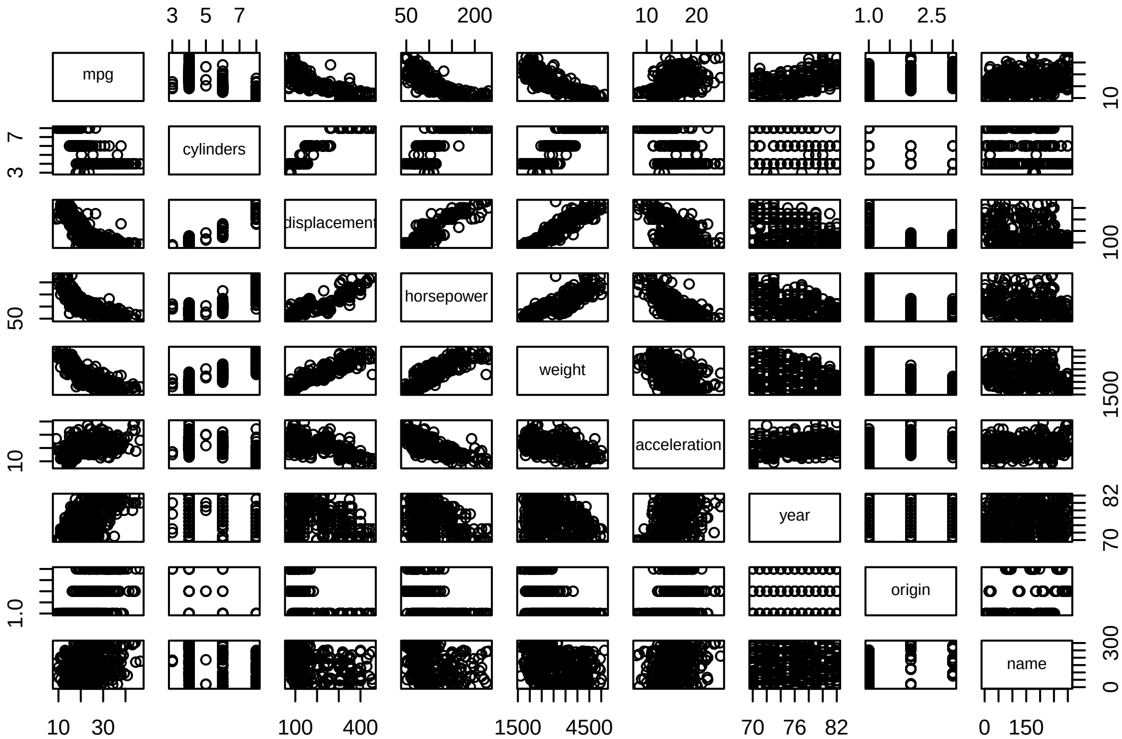


- In Residuals vs Fitted plot, we can see the U-shape curve, which indicates the data has non-linearity.
- In Scale-location, we can see that the assumption that variance is constant through examples is liked to be violated.

9

(a)

```
plot(Auto)
```



(b)

```r
cor(subset(Auto, select = -name))
```

```
##                     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
lm.fit <- lm(mpg ~ .-name, data = Auto)
summary(lm.fit)
```

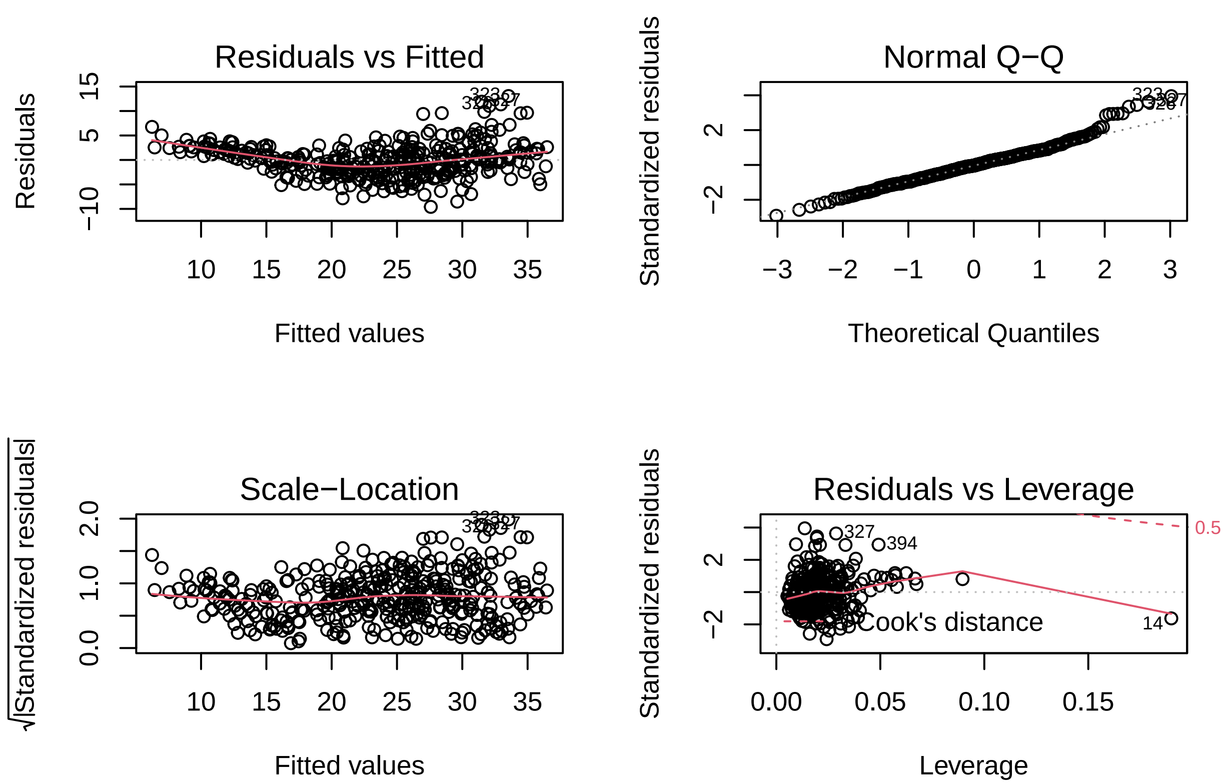
```
##
## Call:
## lm(formula = mpg ~ . - name, data = Auto)
##
```

```
## 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
```

- Looking at F-statistics, there is a relationship between the predictors and the response.
- R^2 statistics is 0.8215, so the linear model explains the relationship.
- displacement, weight, year, and origin have a statistically significant relationship to the response.
- the positive coefficient of `year` variable suggests newer cars are more effective.

(d)

```r
par(mfrow = c(2, 2))
plot(lm.fit)
```



- In Residuals vs Fitted, you can see a slight non-linear trend.
- There is an observation with high leverage.

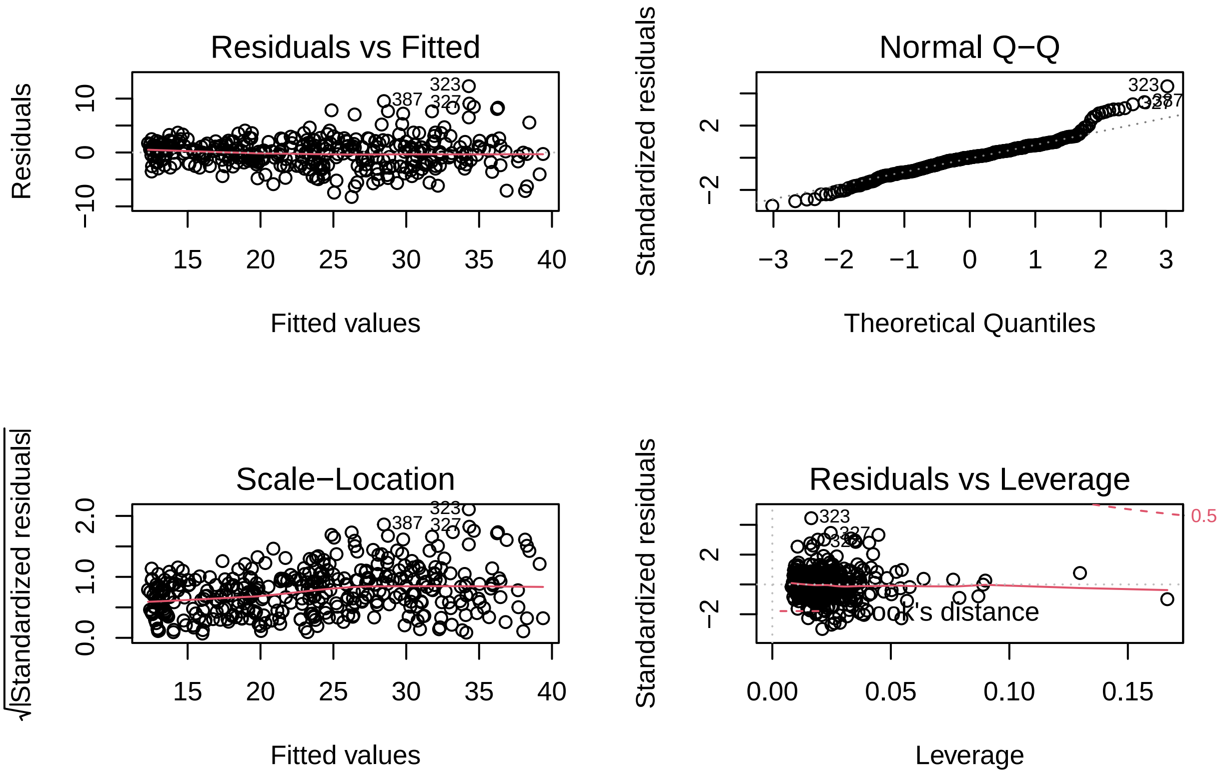
(e)

(f) We applied X^2 transformations to the four predictors.

```
lm.fit2 <- lm(mpg ~ origin + I(origin^2) + year + I(year^2) + weight + I(weight^2) + horsepower + I(hors
summary(lm.fit2)
```

```
##
## Call:
## lm(formula = mpg ~ origin + I(origin^2) + year + I(year^2) +
##     weight + I(weight^2) + horsepower + I(horsepower^2), data = Auto)
##
## Residuals:
##    Min      1Q Median     3Q    Max
## -8.255 -1.674  0.060  1.452 12.305
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      4.081e+02  6.969e+01   5.856 1.02e-08 ***
## origin           3.869e+00  1.602e+00   2.415   0.0162 *
## I(origin^2)     -7.945e-01  4.026e-01  -1.973   0.0492 *
## year            -1.001e+01  1.840e+00  -5.439 9.56e-08 ***
## I(year^2)        7.098e-02  1.209e-02   5.870 9.43e-09 ***
## weight          -1.502e-02  1.764e-03  -8.519 3.74e-16 ***
## I(weight^2)      1.681e-06  2.594e-07   6.479 2.84e-10 ***
## horsepower      -1.420e-01  2.870e-02  -4.949 1.12e-06 ***
## I(horsepower^2)  4.063e-04  1.015e-04   4.002 7.54e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.793 on 383 degrees of freedom
## Multiple R-squared:  0.8746, Adjusted R-squared:  0.872
## F-statistic: 333.9 on 8 and 383 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2, 2))
plot(lm.fit2)
```



*

We have higher R^2 Statistic even though some predictors are discarded. * In Residuals vs Fitted values plot, the non-linear trend is gone. * In scale-Location plot, it looks like the variance is constant.

10

(a)

```
lm.fit <- lm(Sales ~ Price + Urban + US, data = Carseats)
```

(b)

```
summary(lm.fit)
```

```
##
## Call:
## lm(formula = Sales ~ Price + Urban + US, data = Carseats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9206 -1.6220 -0.0564  1.5786  7.0581
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.043469   0.651012  20.036  < 2e-16 ***
## Price       -0.054459   0.005242 -10.389  < 2e-16 ***
## UrbanYes    -0.021916   0.271650  -0.081    0.936
## USYes        1.200573   0.259042   4.635 4.86e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.472 on 396 degrees of freedom
```

```
## Multiple R-squared:  0.2393, Adjusted R-squared:  0.2335
## F-statistic: 41.52 on 3 and 396 DF,  p-value: < 2.2e-16
```

Looking at the p-values of the coefficients, whether a store is in an urban location or not does not have an effect on sales while whether a store is in the U.S. or not and the price have association with the sales. If the price goes up, the sales go down. If a store is in the U.S., the sales go up.

(c)

$$Sales = 13.043469 - 0.054459 * Price - -0.02191 * UrbanYes + 1.200573 * USYes$$

(d) Price and US

(e)

```
small.fit <- lm(Sales ~ Price + US, data = Carseats)
summary(small.fit)
```

```
##
## Call:
## lm(formula = Sales ~ Price + US, data = Carseats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9269 -1.6286 -0.0574  1.5766  7.0515
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.03079    0.63098  20.652  < 2e-16 ***
## Price       -0.05448    0.00523 -10.416  < 2e-16 ***
## USYes        1.19964    0.25846   4.641 4.71e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.469 on 397 degrees of freedom
## Multiple R-squared:  0.2393, Adjusted R-squared:  0.2354
## F-statistic: 62.43 on 2 and 397 DF,  p-value: < 2.2e-16
```

(f)

The smaller model has higher adjusted R^2 statistic, so the smaller one fits to the data better.

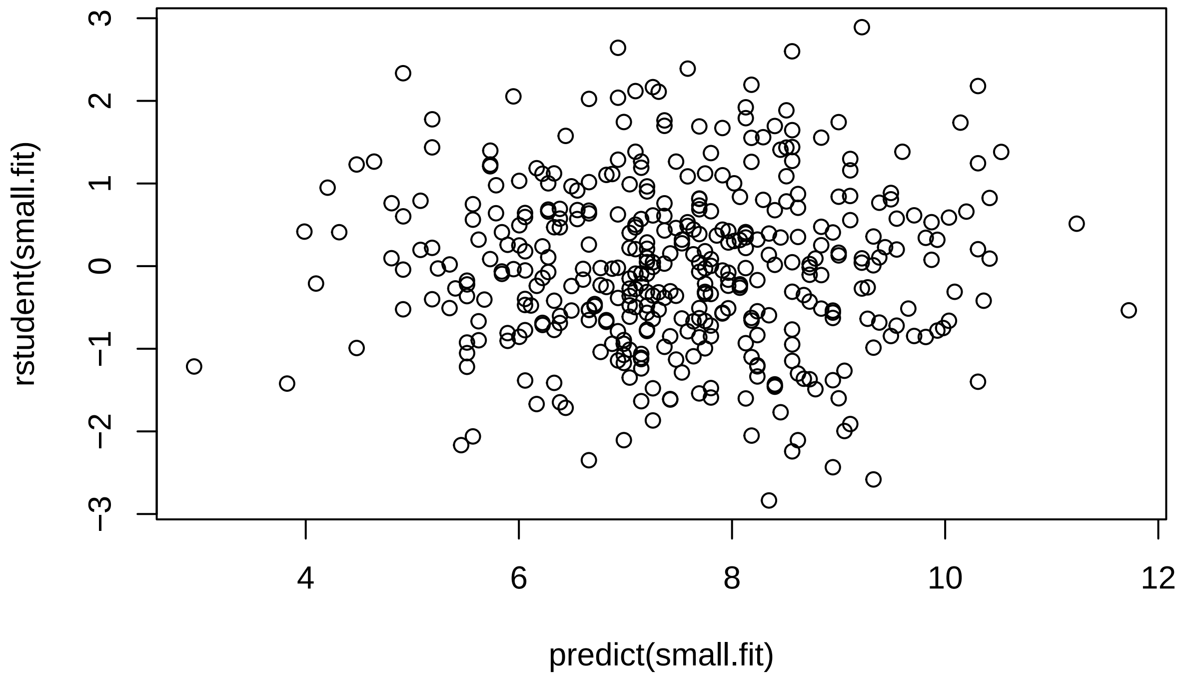
(g)

```
confint(small.fit)
```

```
##                   2.5 %      97.5 %
## (Intercept) 11.79032020 14.27126531
## Price       -0.06475984 -0.04419543
## USYes        0.69151957  1.70776632
```

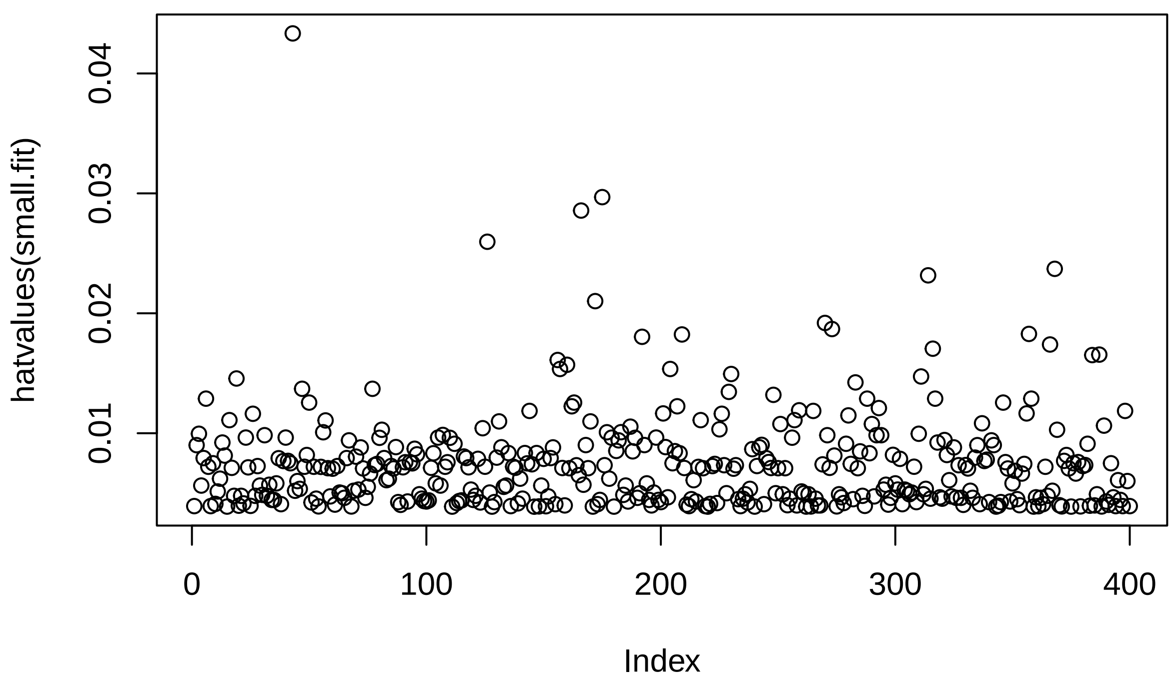
(h)

```
plot(predict(small.fit), rstudent(small.fit))
```



Studentized residuals of all observations are between -3 and 3. Hence, there is no evidence that there is an outlier.

```
plot(hatvalues(small.fit))
```



The average leverage for all the observations is always $(p+1)/n$. In this case, $3/400 = 0.075$. So, an observation with higher leverage statistic than 0.075 might be a high leverage observation. There is an observation with leverage statistic of around 0.04. Hence, we can say that there is a high leverage observation.

11

```r
set.seed(1)
x <- rnorm(100)
y <- 2 * x + rnorm(100)
```

(a)

```r
lm.fit <- lm(y ~ x - 1)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = y ~ x - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9154 -0.6472 -0.1771  0.5056  2.3109
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## x   1.9939     0.1065   18.73   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9586 on 99 degrees of freedom
## Multiple R-squared:  0.7798, Adjusted R-squared:  0.7776
## F-statistic: 350.7 on 1 and 99 DF,  p-value: < 2.2e-16
```

The t-statistic of the coefficient is so high (the p-value is so low) that the null hypothesis that the coefficient is zero.

(b)

```r
lm.fit2 <- lm(x ~ y - 1)
summary(lm.fit2)
```

```
##
## Call:
## lm(formula = x ~ y - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8699 -0.2368  0.1030  0.2858  0.8938
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## y  0.39111    0.02089   18.73   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4246 on 99 degrees of freedom
## Multiple R-squared:  0.7798, Adjusted R-squared:  0.7776
## F-statistic: 350.7 on 1 and 99 DF,  p-value: < 2.2e-16
```

The t-statistc and p-value is identical to the regression of Y on X.

(d)

The model is

$$y = \beta x + \epsilon \quad \epsilon \sim \mathcal{N}(0, \sigma)$$

$\widehat{\beta} = \frac{\sum x_i y_i}{\sum x_i^2}$, so

$$Var[\widehat{\beta}] = \frac{\sum x_i^2 \sigma^2}{\left(\sum x_i^2\right)} = \frac{\sigma^2}{\sum x_i^2}$$

$\widehat{\sigma} = \frac{RSS}{n-1} = \frac{\sum \left(y_i - \widehat{\beta} x_i\right)}{n-1}$.

Therefore, $SE[\beta] = \sqrt{\frac{\sum \left(y_i - \widehat{\beta} x_i\right)}{(n-1)\left(\sum x_i^2\right)}}$.

Next, we calculate the t-statistic $\widehat{\beta}/SE[\widehat{\beta}]$. Substituting $\widehat{\beta}$ and $E[\widehat{\beta}]$ with the above and simplifying, we have the answer.

```
t <- sqrt(length(x) - 1) * sum(x * y) / sqrt(sum(x * x) * sum(y * y) - sum(x * y) ** 2)
print(t)
```

```
## [1] 18.72593
```

(e) The t-statistic is symmetric, so the t-statistic of the regression of Y on X is the same as that of the regression of X on Y.

(f)

```
lm.fit2 <- lm(y ~ x)
summary(lm.fit2)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8768 -0.6138 -0.1395  0.5394  2.3462
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.03769    0.09699  -0.389    0.698
## x            1.99894    0.10773  18.556   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9628 on 98 degrees of freedom
## Multiple R-squared:  0.7784, Adjusted R-squared:  0.7762
## F-statistic: 344.3 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
lm.fit3 <- lm(x ~ y)
summary(lm.fit3)
```

```
##
## Call:
## lm(formula = x ~ y)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.90848 -0.28101  0.06274  0.24570  0.85736
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.03880    0.04266    0.91    0.365
## y            0.38942    0.02099   18.56   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4249 on 98 degrees of freedom
## Multiple R-squared:  0.7784, Adjusted R-squared:  0.7762
## F-statistic: 344.3 on 1 and 98 DF,  p-value: < 2.2e-16
```

12

(a) If $\sum x_i^2 = \sum y_i^2$, the estimated coefficient is the same.

(b)

```
n <- 100
set.seed(1)
x <- rnorm(100)
y <- 3 * x + rnorm(100, 0, 0.1)
```

```
lm.fit <- lm(y ~ x -1)
coef(lm.fit)
```

```
##        x
## 2.999388
```

```
lm.fit2 <- lm(x ~ y - 1)
coef(lm.fit2)
```

```
##        y
## 0.332986
```

(c)

```
set.seed(1)
x <- rnorm(100)
y <- sample(x, 100)
```

```
lm.fit <- lm(y ~ x - 1)
coef(lm.fit)
```

```
##           x
## -0.07767695
```

```
lm.fit2 <- lm(y ~ x - 1)
coef(lm.fit2)
```

```
##           x
## -0.07767695
```

13

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

(a)

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

(b)

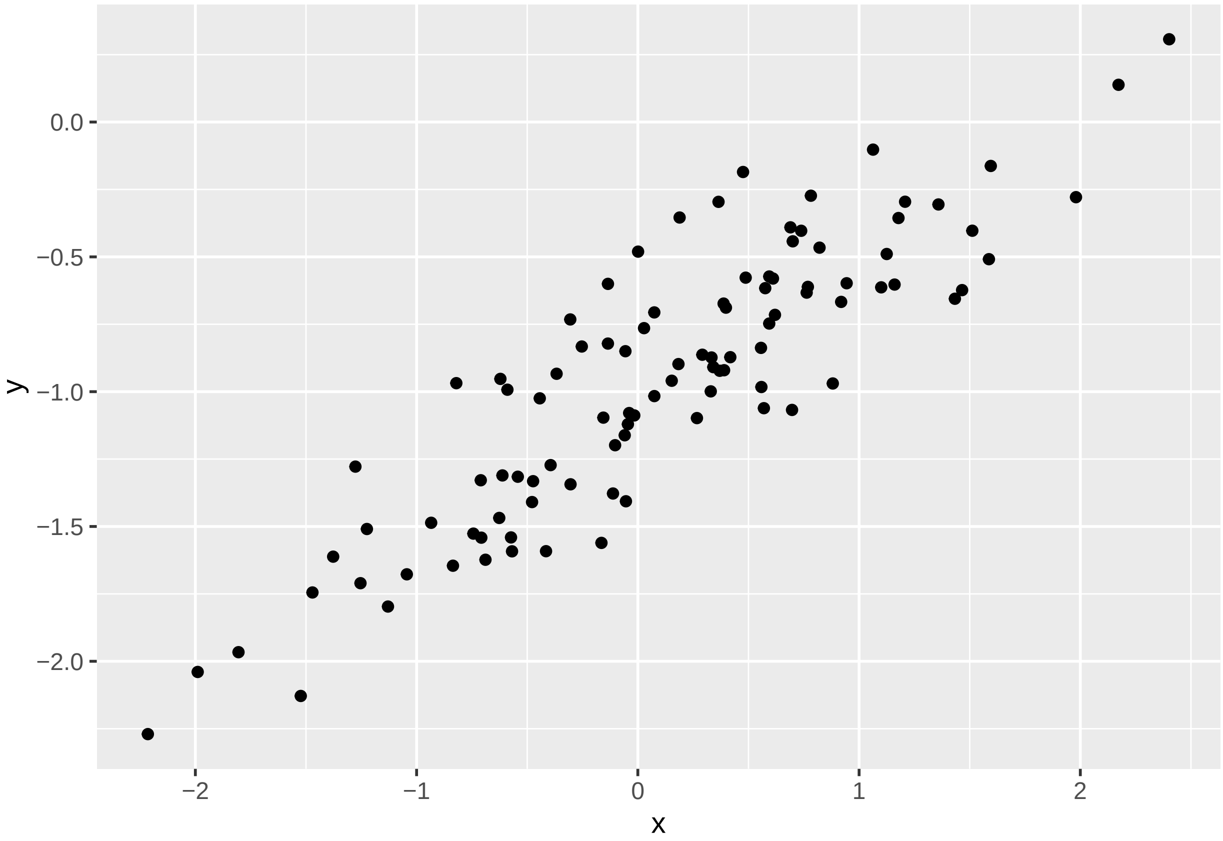
```
eps <- rnorm(100, 0, 0.25)
```

(c)

```
y <- -1 + 0.5 * x + eps
```

(d)

```
ggplot(mapping = aes(x = x, y = y)) +
  geom_point()
```



There is a linear relationship between y and x.

(e)

```
lm.fit <- lm(y ~ x)
summary(lm.fit)
```

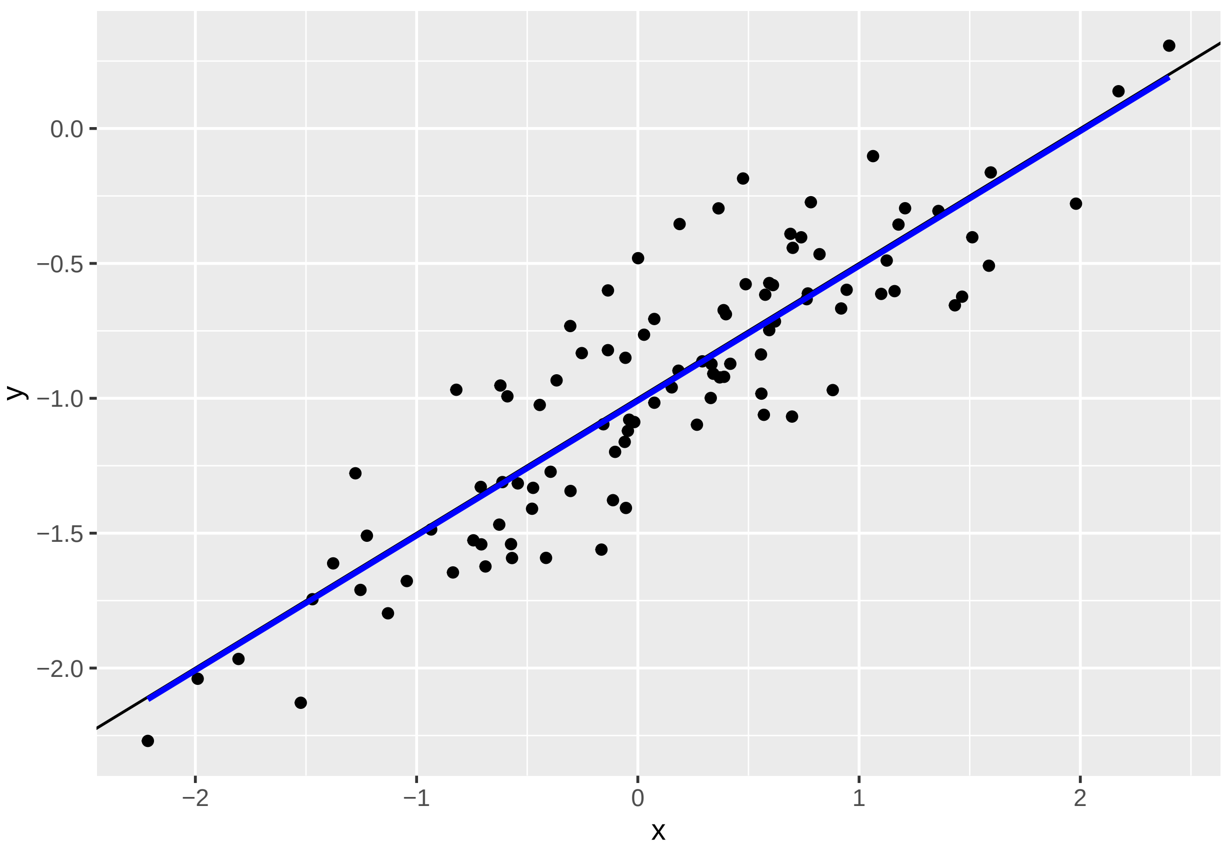
```
##
## Call:
## lm(formula = y ~ x)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46921 -0.15344 -0.03487  0.13485  0.58654
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.00942    0.02425  -41.63   <2e-16 ***
## x            0.49973    0.02693   18.56   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2407 on 98 degrees of freedom
## Multiple R-squared:  0.7784, Adjusted R-squared:  0.7762
## F-statistic: 344.3 on 1 and 98 DF,  p-value: < 2.2e-16
```

Estimated coefficients are very close to the true ones.

(f)

```
ggplot() +
  geom_point(mapping = aes(x = x, y = y)) +
  geom_abline(slope = 0.5, intercept = -1, show.legend =  TRUE) +
  geom_smooth(mapping = aes(x = x, y = y), method = 'lm', formula = y ~ x, color = "blue", se = FALSE)
```



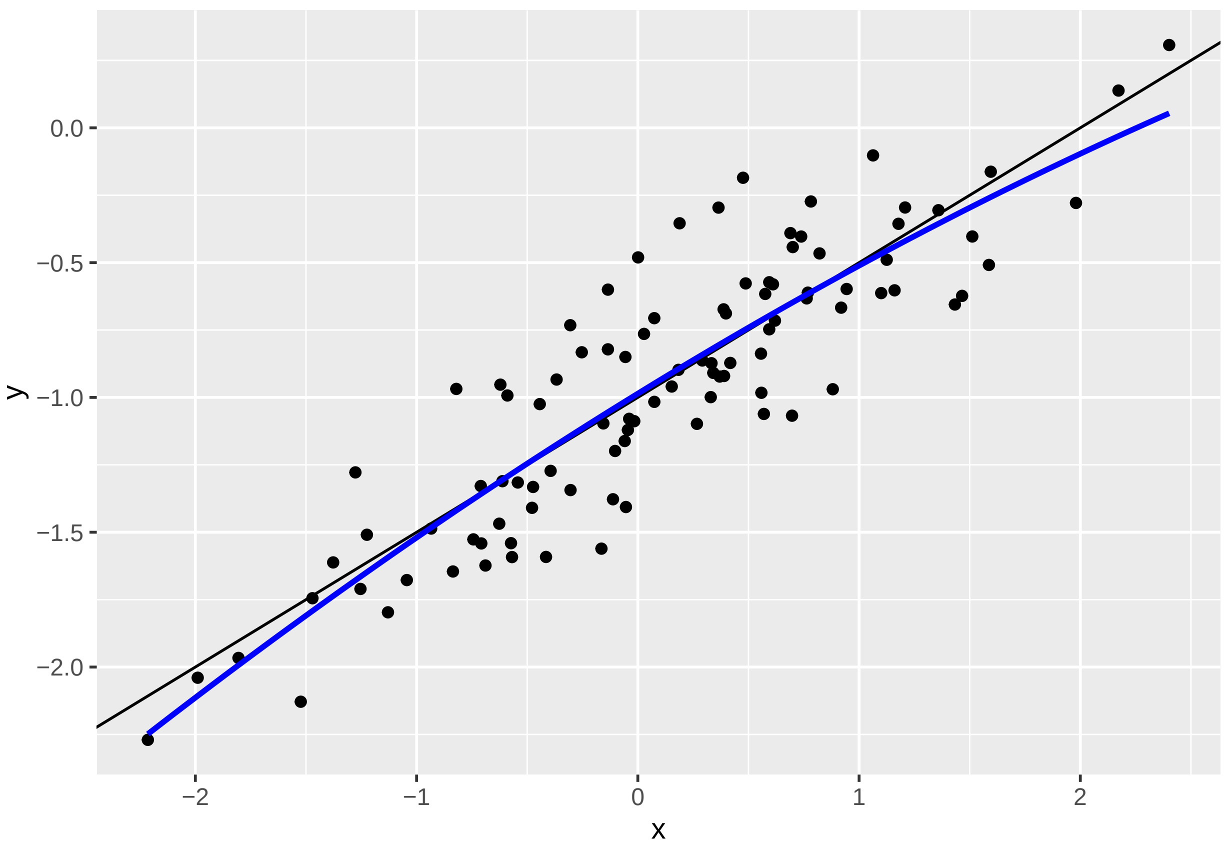
(g)

```
lm.fit2 <- lm(y ~ poly(x, 2))
summary(lm.fit2)
```

```
##
```

```
## Call:
## lm(formula = y ~ poly(x, 2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.4913 -0.1563 -0.0322  0.1451  0.5675
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.95501    0.02395 -39.874   <2e-16 ***
## poly(x, 2)1  4.46612    0.23951  18.647   <2e-16 ***
## poly(x, 2)2 -0.33602    0.23951  -1.403    0.164
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2395 on 97 degrees of freedom
## Multiple R-squared:  0.7828, Adjusted R-squared:  0.7784
## F-statistic: 174.8 on 2 and 97 DF,  p-value: < 2.2e-16
```

```r
ggplot() +
  geom_point(mapping = aes(x = x, y = y)) +
  geom_abline(slope = 0.5, intercept = -1, show.legend =  TRUE) +
  geom_smooth(mapping = aes(x = x, y = y), method = 'lm', formula = y ~ x + I(x^2), color = "blue", se =
```



Looking at the graph, the previous model looks better, but looking at the summary info. the quadratic model looks better because the X^2 coefficient has low p-value and adjusted R^2 statistic is higher than the previous model.

(h)

```r
for (s in c(0.20, 0.15, 0.10, 0.05, 0.01)) {
  set.seed(1)
  x <- rnorm(100)
  eps <- rnorm(100, 0, s)
  y <- -1 + 0.5 * x + eps
  lm.fit <- lm(y ~ x)
  print(summary(lm.fit))
}
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.3754 -0.1227 -0.0279  0.1079  0.4692
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.00754    0.01940  -51.94   <2e-16 ***
## x            0.49979    0.02155   23.20   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1926 on 98 degrees of freedom
## Multiple R-squared:  0.8459, Adjusted R-squared:  0.8444
## F-statistic: 538.1 on 1 and 98 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.28153 -0.09206 -0.02092  0.08091  0.35193
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.00565    0.01455  -69.13   <2e-16 ***
## x            0.49984    0.01616   30.93   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1444 on 98 degrees of freedom
## Multiple R-squared:  0.9071, Adjusted R-squared:  0.9061
## F-statistic: 956.8 on 1 and 98 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -0.18768 -0.06138 -0.01395  0.05394  0.23462
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.003769   0.009699  -103.5   <2e-16 ***
## x            0.499894   0.010773    46.4   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09628 on 98 degrees of freedom
## Multiple R-squared:  0.9565, Adjusted R-squared:  0.956
## F-statistic:  2153 on 1 and 98 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.093842 -0.030688 -0.006975  0.026970  0.117309
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.001885   0.004849 -206.60   <2e-16 ***
## x            0.499947   0.005386   92.82   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04814 on 98 degrees of freedom
## Multiple R-squared:  0.9888, Adjusted R-squared:  0.9886
## F-statistic:  8615 on 1 and 98 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.018768 -0.006138 -0.001395  0.005394  0.023462
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.0003769  0.0009699 -1031.5   <2e-16 ***
## x            0.4999894  0.0010773   464.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.009628 on 98 degrees of freedom
## Multiple R-squared:  0.9995, Adjusted R-squared:  0.9995
## F-statistic: 2.154e+05 on 1 and 98 DF,  p-value: < 2.2e-16
```

When the noise gets smaller, the model fitting gets better. The estimated standard errors of the coefficients get smaller, too.

(i)

```r
for (s in c(0.25, 0.30, 0.35, 0.40, 0.45)) {
  set.seed(1)
  x <- rnorm(100)
  eps <- rnorm(100, 0, s)
  y <- -1 + 0.5 * x + eps
  lm.fit <- lm(y ~ x)
  print(summary(lm.fit))
}
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46921 -0.15344 -0.03487  0.13485  0.58654
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.00942    0.02425  -41.63   <2e-16 ***
## x            0.49973    0.02693   18.56   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2407 on 98 degrees of freedom
## Multiple R-squared:  0.7784, Adjusted R-squared:  0.7762
## F-statistic: 344.3 on 1 and 98 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.56305 -0.18413 -0.04185  0.16182  0.70385
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.01131    0.02910  -34.76   <2e-16 ***
## x            0.49968    0.03232   15.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2888 on 98 degrees of freedom
## Multiple R-squared:  0.7092, Adjusted R-squared:  0.7063
## F-statistic: 239.1 on 1 and 98 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.65689 -0.21482 -0.04882  0.18879  0.82116
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.01319    0.03395  -29.85   <2e-16 ***
## x            0.49963    0.03770   13.25   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.337 on 98 degrees of freedom
## Multiple R-squared:  0.6418, Adjusted R-squared:  0.6381
## F-statistic: 175.6 on 1 and 98 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7507 -0.2455 -0.0558  0.2158  0.9385
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.01508    0.03879  -26.16   <2e-16 ***
## x            0.49958    0.04309   11.59   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3851 on 98 degrees of freedom
## Multiple R-squared:  0.5783, Adjusted R-squared:  0.574
## F-statistic: 134.4 on 1 and 98 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.84458 -0.27619 -0.06277  0.24273  1.05578
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.01696    0.04364   -23.3   <2e-16 ***
## x            0.49952    0.04848    10.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4332 on 98 degrees of freedom
## Multiple R-squared:   0.52,  Adjusted R-squared:  0.5151
## F-statistic: 106.2 on 1 and 98 DF,  p-value: < 2.2e-16
```

The fitting gets worse when the noise gets bigger.

(j)

```r
x <- rnorm(100)
eps <- rnorm(100, 0, 0.25)
y <- -1 + 0.5 * x + eps
lm.fit <- lm(y ~ x)
confint(lm.fit)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.0370526 -0.9387224
## x            0.4787987  0.5743105
```

```r
x <- rnorm(100)
eps <- rnorm(100, 0, 0.05)
y <- -1 + 0.5 * x + eps
lm.fit <- lm(y ~ x)
confint(lm.fit)
```

```
##                 2.5 %     97.5 %
## (Intercept) -1.011974 -0.9927714
## x            0.488007  0.5044982
```

```r
x <- rnorm(100)
eps <- rnorm(100, 0, 0.5)
y <- -1 + 0.5 * x + eps
lm.fit <- lm(y ~ x)
confint(lm.fit)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.1006008 -0.8790924
## x            0.4489768  0.6514931
```