Lab 3: Linear Regression II

Linear Model Selection & Problems with Linear Regression

ACTL3142 and ACTL5110

Questions

Conceptual Questions

1. ⋆ (ISLR2, Q6.1) We perform best subset, forward stepwise, and backward stepwise selection on a single data set. For each approach, we obtain $p+1$ models, containing $0, 1, 2, \ldots, p$ predictors. Explain your answers:

   a. Which of the three models with k predictors has the smallest *training* RSS?

   b. Which of the three models with k predictors has the smallest *test* RSS?

   c. True or False:

      i. The predictors in the k-variable model identified by forward stepwise are a subset of the predictors in the $(k+1)$-variable model identified by forward stepwise selection.

      ii. The predictors in the k-variable model identified by backward stepwise are a subset of the predictors in the $(k+1)$- variable model identified by backward stepwise selection

      iii. The predictors in the k-variable model identified by backward stepwise are a subset of the predictors in the $(k+1)$- variable model identified by forward stepwise selection.

      iv. The predictors in the k-variable model identified by forward stepwise are a subset of the predictors in the $(k+1)$-variable model identified by backward stepwise selection.

      v. The predictors in the k-variable model identified by best subset are a subset of the predictors in the $(k+1)$-variable model identified by best subset selection.

   Solution

Applied Questions

1. $\star$ (ISLR2, Q6.8) In this exercise, we will generate simulated data, and will then use this data to perform best subset selection.

   a. Use the `rnorm()` function to generate a predictor X of length $n = 100$, as well as a noise vector ϵ of length $n = 100$.

   b. Generate a response vector Y of length $n = 100$ according to the model

   $$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon,$$

   where β_0, β_1, β_2, and β_3 are constants of your choice.

   c. Use the `regsubsets()` function to perform best subset selection in order to choose the best model containing the predictors $X, X^2, \ldots, X^{10}$. What is the best model obtained according to C_p, BIC, and adjusted R^2? Show some plots to provide evidence for your answer, and report the coefficients of the best model obtained. Note you will need to use the `data.frame()` function to create a single data set containing both X and Y.

   d. Repeat (c), using forward stepwise selection and also using backwards stepwise selection. How does your answer compare to the results in (c)?

   e. Now generate a response vector Y according to the model

   $$Y = \beta_0 + \beta_7 X^7 + \epsilon,$$

   and perform best subset selection. Discuss the results obtained.

   Solution

2. $\star$ (ISLR2, Q3.10) This question should be answered using the `Carseats` data set.

   a. Fit a multiple regression model to predict `Sales` using `Price`, `Urban`, and `US`.

   b. Provide an interpretation of each coefficient in the model. Be careful—some of the variables in the model are qualitative!

   c. Write out the model in equation form, being careful to handle the qualitative variables properly.

   d. For which of the predictors can you reject the null hypothesis $H_0 : \beta_j = 0$?

   e. On the basis of your response to the previous question, fit a smaller model that only uses the predictors for which there is evidence of association with the outcome.

   f. How well do the models in (a) and (e) fit the data?

   g. Using the model from (e), obtain 95% confidence intervals for the coefficient(s).

h. Is there evidence of outliers or high leverage observations in the model from (e)?

Solution

3. (ISLR2, Q3.13) In this exercise you will create some simulated data and will fit simple linear regression models to it. Make sure to use set.seed(1) prior to starting part (a) to ensure consistent results.

   a. Using the `rnorm()` function, create a vector, `x`, containing 100 observations drawn from a $N(0,1)$ distribution. This represents a feature, X.

   b. Using the `rnorm()` function, create a vector, `eps`, containing 100 observations drawn from a $N(0,0.25)$ distribution—a normal distribution with mean zero and variance 0.25.

   c. Using `x` and `eps`, generate a vector y according to the model

$$Y = -1 + 0.5X + \epsilon. \tag{1}$$

   What is the length of the vector `y`? What are the values of β_0 and β_1 in this linear model?

   d. Create a scatterplot displaying the relationship between `x` and `y`. Comment on what you observe.

   e. Fit a least squares linear model to predict `y` using `x`. Comment on the model obtained. How do $\hat{\beta}_0$ and $\hat{\beta}_1$ compare to β_0 and β_1?

   f. Display the least squares line on the scatterplot obtained in (d). Draw the population regression line on the plot, in a different color. Use the `legend()` command to create an appropriate legend.

   g. Now fit a polynomial regression model that predicts `y` using `x` and `x^2` (X^2). Is there evidence that the quadratic term improves the model fit? Explain your answer.

   h. Repeat (a)–(f) after modifying the data generation process in such a way that there is *less* noise in the data. The model Equation 1 should remain the same. You can do this by decreasing the variance of the normal distribution used to generate the error term ϵ in (b). Describe your results.

   i. Repeat (a)–(f) after modifying the data generation process in such a way that there is *more* noise in the data. The model Equation 1 should remain the same. You can do this by increasing the variance of the normal distribution used to generate the error term ϵ in (b). Describe your results.

   j. What are the confidence intervals for β_0 and β_1 based on the original data set, the noisier data set, and the less noisy data set? Comment on your results.

Solution

4. ⋆ (ISLR2, Q3.14) This problem focuses on the *collinearity* problem.

   a. Perform the following commands in `R`:

   ```
   set.seed(1)
   x1 <- runif(100)
   x2 <- 0.5 * x1 + rnorm(100) / 10
   y <- 2 + 2 * x1 + 0.3 * x2 + rnorm(100)
   ```

   The last line corresponds to creating a linear model in which `y` is a function of `x1` and `x2`. Write out the form of the linear model. What are the regression coefficients?

   b. What is the correlation between `x1` and `x2`? Create a scatterplot displaying the relationship between the variables.

   c. Using this data, fit a least squares regression to predict `y` using `x1` and `xx2`. Describe the results obtained. What are $\hat{\beta}_0$, $\hat{\beta}_1$, and $\hat{\beta}_2$? How do these relate to the true β_0, β_1, and β_2? Can you reject the null hypothesis $H_0 : \beta_1 = 0$? How about the null hypothesis$H_0 : \beta_2 = 0$?

   d. Now fit a least squares regression to predict `y` using only `x1`. Comment on your results. Can you reject the null hypothesis $H_0 : \beta_1 = 0$?

   e. Now fit a least squares regression to predict `y` using only `x1`. Comment on your results. Can you reject the null hypothesis $H_0 : \beta_1 = 0$?

   f. Do the results obtained in (c)–(e) contradict each other? Explain your answer.

   g. Now suppose we obtain one additional observation, which was unfortunately mismeasured.

   ```
   set.seed(1)
   x1 <- c(x1, 0.1)
   x2 <- c(x2, 0.8)
   y <- c(y, 6)
   ```

   Re-fit the linear models from (c) to (e) using this new data. What effect does this new observation have on the each of the models? In each model, is this observation an outlier? A high-leverage point? Both? Explain your answers.

Solution

Solutions

Conceptual Questions

1. a. Best-subset selection. It will always pick the size-k model which will best fit the data, in other words, has the lowest RSS. The other two will select the model which minimises the RSS, given their selections for the $k \pm 1$ model.

   b. It's most likely going to be best-subset, since for a given degree of flexibility, a smaller training RSS usually leads to a smaller test RSS, but forward and backward-stepwise could luck out and find a model which works better on the test data.

   c. i. True. Forward stepwise adds a predictor to the best-found model of the previous step. So it will add one predictor to the size-k to get a size-$k+1$ model.

      ii. True. Backward stepwise removes a predictor from the best-found model of the previous step. So, it will remove one predictor from the size-$k+1$ model to obtain the "best" size-k model.

      iii. False. Forward and backward stepwise can diverge in what they consider the "best" model. This is due to their different starting positions.

      iv. False. Same reasoning as iii.

      v. False. Best-subset considers all possible selections of k parameters to find the best size-k model. It does not consider what the best $k-1$ or $k+1$ model was. Hence, the size-k model's predictors could be entirely different to those of the size-$k+1$'s.

Applied Questions

1. a.
```
set.seed(1)
X <- rnorm(100, 10, 3)
noise <- rnorm(100, 0, 2)
```

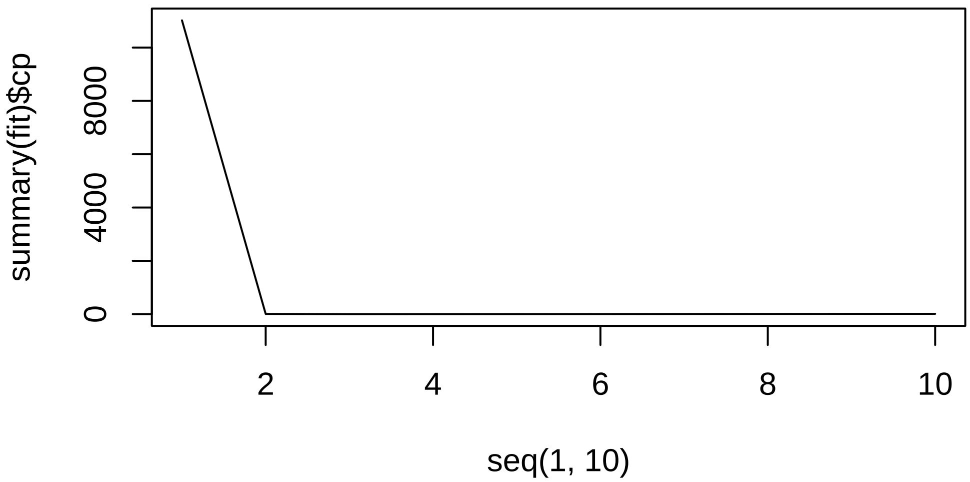
   b.
```
Y <- 4 + 3 * X + 2 * X^2 + X^3 + noise
```

   c.
```
myData <- data.frame(
  Y = Y, X1 = X, X2 = X^2, X3 = X^3, X4 = X^4, X5 = X^5,
  X6 = X^6, X7 = X^7, X8 = X^8, X9 = X^9, X10 = X^10
)
library(leaps)
fit <- regsubsets(Y ~ ., data = myData, nvmax = 10)
summary(fit)
```

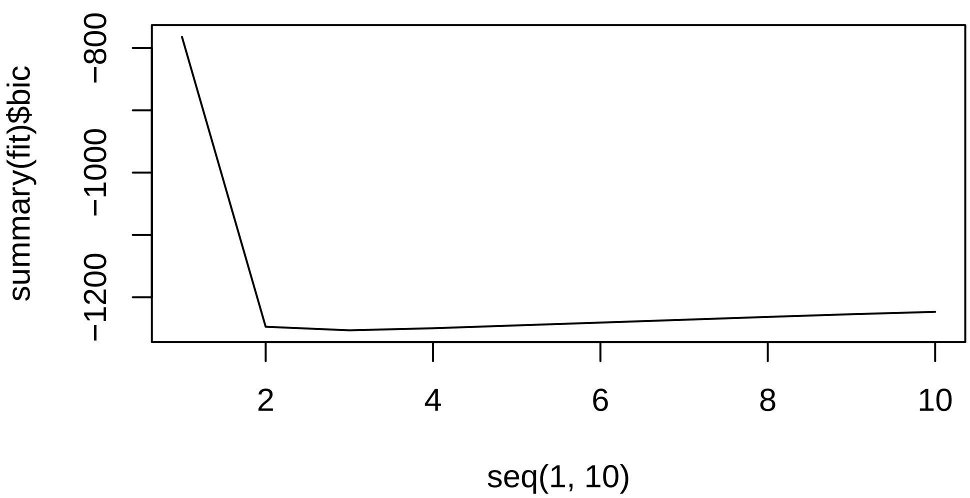
```
Subset selection object
Call: regsubsets.formula(Y ~ ., data = myData, nvmax = 10)
10 Variables  (and intercept)
    Forced in Forced out
X1      FALSE      FALSE
X2      FALSE      FALSE
X3      FALSE      FALSE
X4      FALSE      FALSE
X5      FALSE      FALSE
X6      FALSE      FALSE
X7      FALSE      FALSE
X8      FALSE      FALSE
X9      FALSE      FALSE
X10     FALSE      FALSE
1 subsets of each size up to 10
Selection Algorithm: exhaustive
          X1  X2  X3  X4  X5  X6  X7  X8  X9  X10
1  ( 1 )  " " " " "*" " " " " " " " " " " " " " "
2  ( 1 )  " " "*" "*" " " " " " " " " " " " " " "
3  ( 1 )  " " "*" "*" " " " " " " " " " " " " "*"
4  ( 1 )  "*" " " "*" " " "*" "*" " " " " " " " "
5  ( 1 )  " " "*" " " "*" "*" "*" "*" " " " " " "
6  ( 1 )  " " "*" "*" "*" " " " " " " "*" "*" "*"
7  ( 1 )  " " "*" " " " " "*" "*" "*" "*" "*" "*"
8  ( 1 )  " " " " "*" "*" "*" "*" "*" "*" "*" "*"
9  ( 1 )  " " "*" "*" "*" "*" "*" "*" "*" "*" "*"
10  ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
```

```
myData <- data.frame(
  Y = Y, X1 = X, X2 = X^2, X3 = X^3, X4 = X^4, X5 = X^5,
  X6 = X^6, X7 = X^7, X8 = X^8, X9 = X^9, X10 = X^10
)
library(leaps)
fit <- regsubsets(Y ~ ., data = myData, nvmax = 10)
```

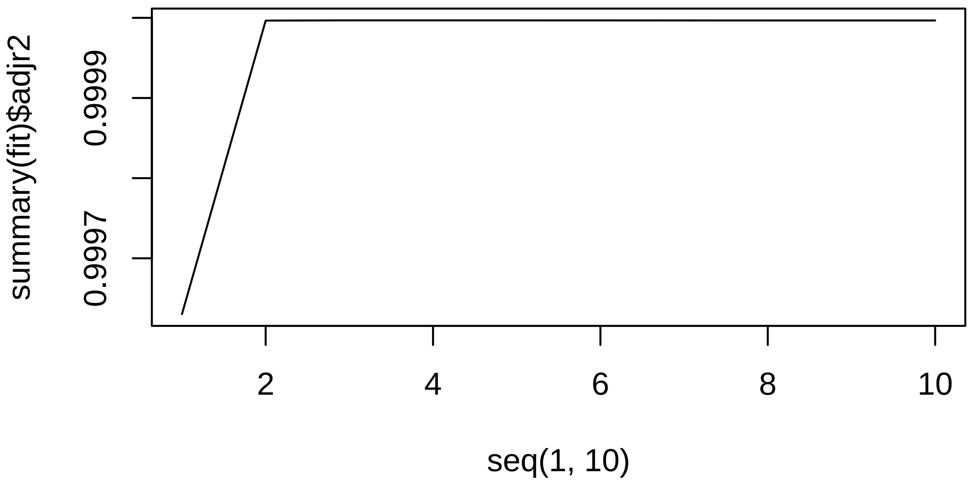
```
plot(seq(1, 10), summary(fit)$cp, type = "l")
```




```r
plot(seq(1, 10), summary(fit)$bic, type = "l")
```




```r
plot(seq(1, 10), summary(fit)$adjr2, type = "l")
```



It looks like all 3 methods suggest that the 3-variable fit is the best.

```
summary(fit)$cp # this should give a better/clearer view
```

```
 [1] 11022.4698919     7.2797918    -0.6713653     0.1801732     2.1193057
 [6]     4.0389022     5.9961670     7.9349196     9.7216590    11.0000000
```

```
summary(fit)
```

```
Subset selection object
Call: regsubsets.formula(Y ~ ., data = myData, nvmax = 10)
10 Variables  (and intercept)
    Forced in Forced out
X1      FALSE      FALSE
X2      FALSE      FALSE
X3      FALSE      FALSE
X4      FALSE      FALSE
X5      FALSE      FALSE
X6      FALSE      FALSE
X7      FALSE      FALSE
X8      FALSE      FALSE
X9      FALSE      FALSE
X10     FALSE      FALSE
1 subsets of each size up to 10
Selection Algorithm: exhaustive
```

```
          X1  X2  X3  X4  X5  X6  X7  X8  X9  X10
1  ( 1 )  " " " " "*" " " " " " " " " " " " " " "
2  ( 1 )  " " "*" "*" " " " " " " " " " " " " " "
3  ( 1 )  " " "*" "*" " " " " " " " " " " " " "*"
4  ( 1 )  "*" " " "*" " " "*" "*" " " " " " " " "
5  ( 1 )  " " "*" " " "*" "*" "*" "*" " " " " " "
6  ( 1 )  " " "*" "*" "*" " " " " " " "*" "*" "*"
7  ( 1 )  " " "*" " " " " "*" "*" "*" "*" "*" "*"
8  ( 1 )  " " " " "*" "*" "*" "*" "*" "*" "*" "*"
9  ( 1 )  " " "*" "*" "*" "*" "*" "*" "*" "*" "*"
10  ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
```

```r
lm(Y ~ X2 + X3 + X10, data = myData)
```

```
Call:
lm(formula = Y ~ X2 + X3 + X10, data = myData)

Coefficients:
(Intercept)           X2           X3          X10
  9.871e+00    2.426e+00    9.818e-01    5.810e-12
```

This final linear fit produces our required coefficients. We note that the model selected includes `X10` instead of `X1` as in the true model. However, this can be attributed to the high amount of noise in the data.

d. Repeat the above with `method = "forward"` and `method = "backward"` in the `regsubsets` function. Alternatively, the step function can also be used here.

```r
attach(myData)
```

```
The following object is masked _by_ .GlobalEnv:

    Y
```

```r
step(lm(Y ~ 1), Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10,
  direction = "forward"
)
```

```
Start:  AIC=1397.35
Y ~ 1

       Df Sum of Sq       RSS     AIC
+ X3    1 114757732     42006  608.04
+ X2    1 112602846   2196892 1003.74
+ X4    1 111991290   2808448 1028.30
```

```
+ X5    1 106159807   8639931 1140.67
+ X1    1 103502796  11296942 1167.49
+ X6    1  98769980  16029758 1202.48
+ X7    1  90912846  23886892 1242.37
+ X8    1  83280410  31519329 1270.09
+ X9    1  76246003  38553735 1290.24
+ X10   1  69964262  44835476 1305.33
<none>              114799738 1397.35

Step:  AIC=608.04
Y ~ X3

       Df Sum of Sq   RSS    AIC
+ X2    1     41623   383 140.19
+ X1    1     41018   988 235.06
+ X4    1     38889  3117 349.94
+ X5    1     36724  5282 402.70
+ X6    1     34461  7545 438.34
+ X7    1     32254  9752 464.01
+ X8    1     30178 11828 483.30
+ X9    1     28269 13737 498.27
+ X10   1     26537 15469 510.14
<none>              42006 608.04

Step:  AIC=140.19
Y ~ X3 + X2

       Df Sum of Sq    RSS    AIC
+ X10   1    37.596 345.04 131.85
+ X9    1    37.224 345.41 131.96
+ X8    1    36.673 345.96 132.12
+ X7    1    35.924 346.71 132.33
+ X6    1    34.960 347.68 132.61
+ X5    1    33.771 348.87 132.95
+ X4    1    32.355 350.28 133.36
+ X1    1    26.801 355.84 134.93
<none>              382.64 140.19

Step:  AIC=131.85
Y ~ X3 + X2 + X10

       Df Sum of Sq    RSS    AIC
<none>              345.04 131.85
```

```
+ X9    1  0.234893 344.81 133.78
+ X1    1  0.201465 344.84 133.79
+ X8    1  0.194667 344.85 133.79
+ X7    1  0.145658 344.90 133.81
+ X6    1  0.091731 344.95 133.82
+ X5    1  0.040675 345.00 133.84
+ X4    1  0.005680 345.04 133.85


Call:
lm(formula = Y ~ X3 + X2 + X10)

Coefficients:
(Intercept)           X3           X2          X10  
  9.871e+00    9.818e-01    2.426e+00    5.810e-12  
```

```r
step(lm(Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10), Y ~ .,
  direction = "backward"
)
```

```
Start:  AIC=143.27
Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10

       Df Sum of Sq    RSS    AIC
- X1    1    2.7265 338.97 142.07
- X2    1    2.8364 339.08 142.11
- X3    1    2.9696 339.21 142.15
- X4    1    3.0773 339.32 142.18
- X5    1    3.1874 339.43 142.21
- X6    1    3.2898 339.53 142.24
- X7    1    3.3841 339.63 142.27
- X8    1    3.4702 339.71 142.29
- X9    1    3.5485 339.79 142.32
- X10   1    3.6194 339.86 142.34
<none>              336.24 143.27

Step:  AIC=142.07
Y ~ X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10

       Df Sum of Sq    RSS    AIC
- X2    1   0.80571 339.78 140.31
- X4    1   1.02949 340.00 140.38
- X3    1   1.03604 340.01 140.38
- X5    1   1.07147 340.04 140.39
```

```
- X6    1   1.10354 340.07 140.40
- X7    1   1.12948 340.10 140.41
- X8    1   1.15180 340.12 140.41
- X9    1   1.17205 340.14 140.42
- X10   1   1.19094 340.16 140.43
<none>              338.97 142.07

Step:  AIC=140.31
Y ~ X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10

       Df Sum of Sq    RSS    AIC
- X10   1    0.5185 340.30 138.46
- X9    1    0.5352 340.31 138.47
- X8    1    0.5652 340.34 138.48
- X7    1    0.6169 340.39 138.49
- X6    1    0.7065 340.48 138.52
- X5    1    0.8712 340.65 138.57
- X4    1    1.2277 341.00 138.67
- X3    1    4.5218 344.30 139.63
<none>              339.78 140.31

Step:  AIC=138.46
Y ~ X3 + X4 + X5 + X6 + X7 + X8 + X9

       Df Sum of Sq    RSS    AIC
- X9    1    0.1097 340.41 136.50
- X8    1    0.2031 340.50 136.52
- X7    1    0.3659 340.66 136.57
- X6    1    0.6572 340.95 136.66
- X5    1    1.2250 341.52 136.82
- X4    1    2.5848 342.88 137.22
<none>              340.30 138.46
- X3    1   21.8581 362.15 142.69

Step:  AIC=136.5
Y ~ X3 + X4 + X5 + X6 + X7 + X8

       Df Sum of Sq    RSS    AIC
<none>              340.41 136.50
- X8    1     7.536 347.94 136.69
- X7    1     9.306 349.71 137.19
- X6    1    11.903 352.31 137.93
- X5    1    16.360 356.76 139.19
```

```
- X4    1    26.328 366.73 141.95
- X3    1   193.910 534.32 179.58


Call:
lm(formula = Y ~ X3 + X4 + X5 + X6 + X7 + X8)

Coefficients:
(Intercept)           X3           X4           X5           X6           X7
  1.157e+01    2.389e+00   -3.353e-01    4.141e-02   -2.789e-03    9.721e-05
         X8
 -1.372e-06
```

Note that the forward direction gives the same model as best-subset selection but the backward direction gives a different model!

e. For ease, we can just set the following and rerun the above code.

```
myData["Y"] <- 10 + 2 * X^7 + noise
```

In this case, it seems that the best-subset algorithm produces the best model to be the 3-predictor model, with `X6`, `X7` and `X8` as predictors. Lasso has forced some predictors to zero but it still retains 5 predictors and has not done well in this case.

2. a.
```
library(ISLR2)
fit <- lm(Sales ~ Price + Urban + US, data = Carseats)
```

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

Price: significant and potentially large. Higher price is associated with lower sales.

Urban: no significant impact on sales (high p-value).

US: effect significant. US stores tend to have higher sales than non-US.

c. `Sales` = 13.043 - 0.0545 `Price` - 0.0219 I(`Urban`=`Yes`) + 1.2006 I(`US`=`Yes`)

d. `Price`, `US`

e.
```r
fit2 <- lm(Sales ~ Price + US, data = Carseats)
summary(fit2)
```

```
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
```

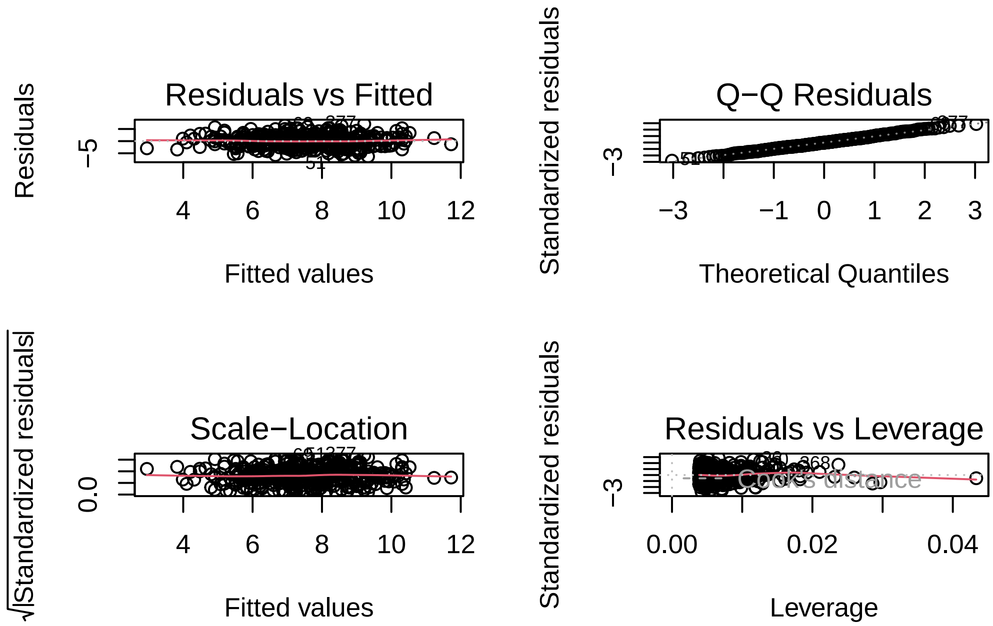
f. About the same: the R2 statistic for both is 0.2393. We prefer (e), though, since it has fewer predictors.

g.
```r
fit2 <- lm(Sales ~ Price + US, data = Carseats)
confint(fit2)
```

```
                  2.5 %      97.5 %
(Intercept) 11.79032020 14.27126531
Price       -0.06475984 -0.04419543
```

```
USYes          0.69151957   1.70776632
```

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



None of the studentised residuals are far outside of the $(-3, 3)$ range, so there are no clear outliers. Some points have a leverage which exceeds $2 \times 3/400 = 0.015$, so there are some high-levered points.

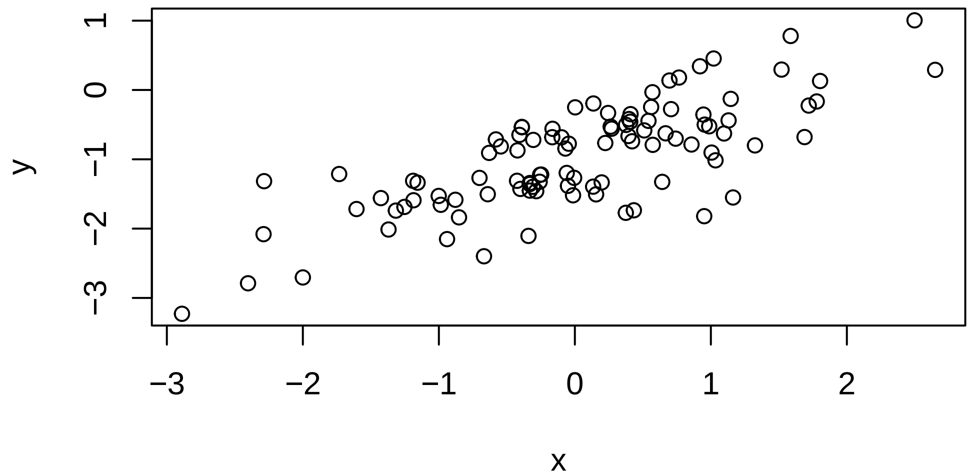
3. a.
```r
x <- rnorm(100)
```

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

c.
```r
y <- -1 + 0.5 * x + eps
```

y is 100 long. $\beta_0 = -1, \beta_0 = 0.5$.

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



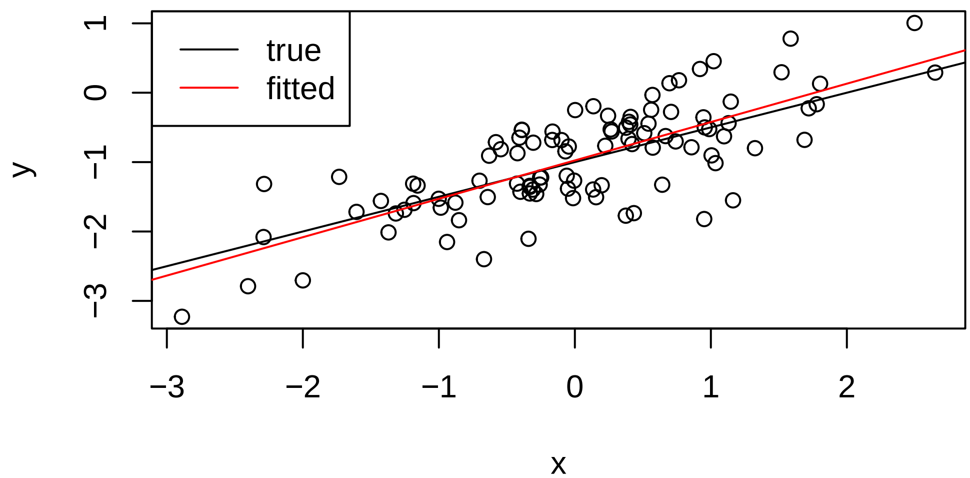
Points are approximately on an increasing straight line.

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

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

Residuals:
     Min       1Q   Median       3Q      Max
-1.37090 -0.28070 -0.00874  0.33987  0.92421

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.97578    0.04955  -19.69   <2e-16 ***
x            0.55311    0.04813   11.49   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4953 on 98 degrees of freedom
Multiple R-squared:  0.574, Adjusted R-squared:  0.5697
F-statistic: 132.1 on 1 and 98 DF,  p-value: < 2.2e-16
```

The β_0 estimate is within 1 standard error, and the β_1 estimate is within 1.1. standard errors.

f.
```
plot(x, y)
abline(-1, 0.5)
abline(fit$coefficients[1], fit$coefficients[2], col = "red")
legend("topleft",
  legend = c("true", "fitted"), lty = 1,
  col = c("black", "red")
)
```



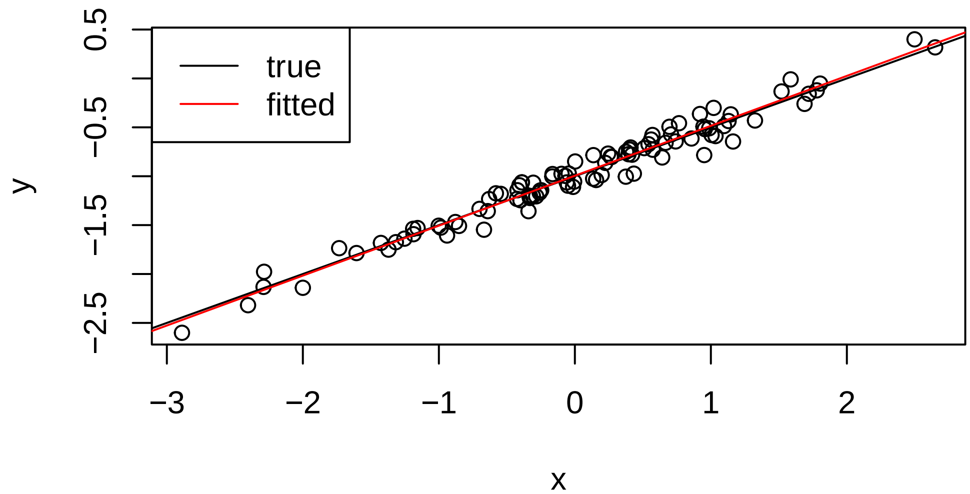
g.
```
fit <- lm(y ~ x + I(x^2))
summary(fit)
```

```

Call:
lm(formula = y ~ x + I(x^2))

Residuals:
     Min       1Q   Median       3Q      Max 
-1.37065 -0.27658 -0.01063  0.32886  0.96560 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.96391    0.05986 -16.104   <2e-16 ***
```

```
x            0.55096    0.04872  11.309   <2e-16 ***
I(x^2)      -0.01114    0.03120  -0.357    0.722
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4975 on 97 degrees of freedom
Multiple R-squared:  0.5746,    Adjusted R-squared:  0.5658
F-statistic: 65.51 on 2 and 97 DF,  p-value: < 2.2e-16
```

No. The adjusted R^2 is higher, indicating the extra term does not have significant explanatory power.

h.
```
set.seed(1)
x <- rnorm(100)
y <- 2 * x + rnorm(100)
x <- rnorm(100)
eps <- rnorm(100, 0, 0.1)
y <- -1 + 0.5 * x + eps
fit2 <- lm(y ~ x)
plot(x, y)
abline(-1, 0.5)
abline(fit2$coefficients[1], fit2$coefficients[2], col = "red")
legend("topleft",
  legend = c("true", "fitted"), lty = 1,
  col = c("black", "red")
)
```



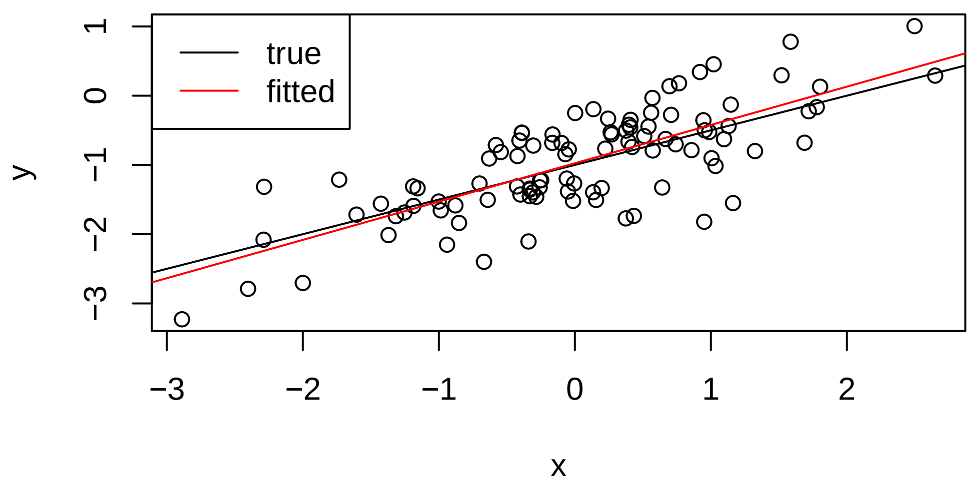
The fitted model is closer to the true model.

i.
```
set.seed(1)
x <- rnorm(100)
y <- 2 * x + rnorm(100)
x <- rnorm(100)
eps <- rnorm(100, 0, 0.5)
y <- -1 + 0.5 * x + eps
fit3 <- lm(y ~ x)
summary(fit)
```

```

Call:
lm(formula = y ~ x + I(x^2))

Residuals:
     Min       1Q   Median       3Q      Max 
-1.37065 -0.27658 -0.01063  0.32886  0.96560 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.96391    0.05986 -16.104   <2e-16 ***
x            0.55096    0.04872  11.309   <2e-16 ***
I(x^2)      -0.01114    0.03120  -0.357    0.722    
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4975 on 97 degrees of freedom
Multiple R-squared:  0.5746,    Adjusted R-squared:  0.5658
F-statistic: 65.51 on 2 and 97 DF,  p-value: < 2.2e-16
```

```
plot(x, y)
abline(-1, 0.5)
abline(fit3$coefficients[1], fit3$coefficients[2], col = "red")
legend("topleft",
  legend = c("true", "fitted"), lty = 1,
  col = c("black", "red")
)
```



The fitted model is further away from the true model.

j. `confint(fit) # noise=0.25`

```
                  2.5 %      97.5 %
(Intercept) -1.08270792 -0.84510882
x            0.45427135  0.64765478
I(x^2)      -0.07306626  0.05079419
```

```
confint(fit2) # noise=0.1
```

```
                  2.5 %     97.5 %
(Intercept) -1.0148210 -0.9754890
x            0.4915195  0.5297242
```

```
confint(fit3) # noise=0.5
```

```
                  2.5 %     97.5 %
(Intercept) -1.0741052 -0.8774448
x            0.4575975  0.6486210
```

The confidence interval is wider when the noise is larger.

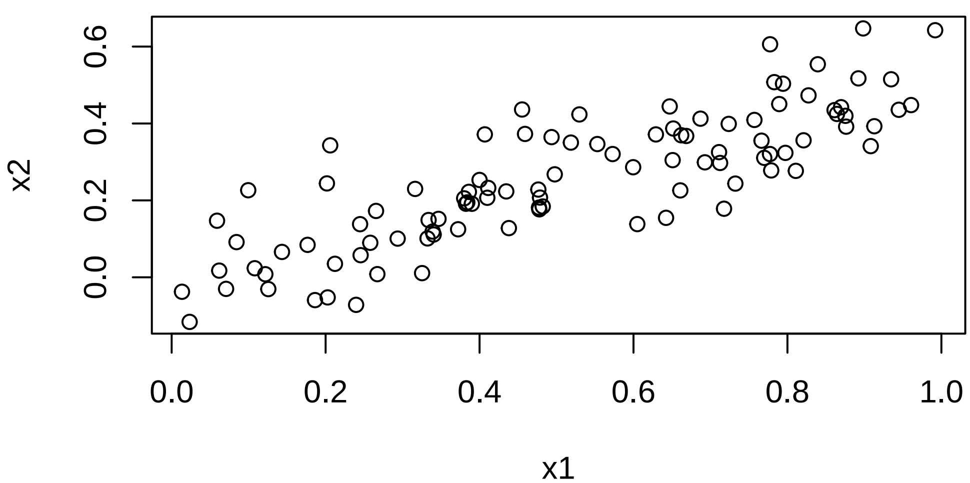
4. a. $\beta_0 = 2, \beta_1 = 2, \beta_2 = 0.3$

```
set.seed(1)
x1 <- runif(100)
x2 <- 0.5 * x1 + rnorm(100) / 10
y <- 2 + 2 * x1 + 0.3 * x2 + rnorm(100)
```

b.
```
cor(x1, x2)
```

```
[1] 0.8351212
```

```
plot(x1, x2)
```



c.
```
summary(lm(y ~ x1 + x2))
```

```
Call:
lm(formula = y ~ x1 + x2)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.8311 -0.7273 -0.0537  0.6338  2.3359 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   2.1305     0.2319   9.188 7.61e-15 ***
x1            1.4396     0.7212   1.996   0.0487 *  
x2            1.0097     1.1337   0.891   0.3754    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.056 on 97 degrees of freedom
Multiple R-squared:  0.2088,	Adjusted R-squared:  0.1925 
F-statistic:  12.8 on 2 and 97 DF,  p-value: 1.164e-05
```

The β_1, β_2 estimates are far off the true values. We can reject the null hypothesis that $\beta_1 = 0$ at the 95

d. `summary(lm(y ~ x1))`

```
Call:
lm(formula = y ~ x1)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.89495 -0.66874 -0.07785  0.59221  2.45560 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   2.1124     0.2307   9.155 8.27e-15 ***
x1            1.9759     0.3963   4.986 2.66e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.055 on 98 degrees of freedom
Multiple R-squared:  0.2024,	Adjusted R-squared:  0.1942 
F-statistic: 24.86 on 1 and 98 DF,  p-value: 2.661e-06
```

All coefficients are significant. The estimate of β_0 is within 1 standard deviation of the true value. The estimate of β_1 is within 1 standard deviation of the true effective value.

e.
```
summary(lm(y ~ x2))
```

```
Call:
lm(formula = y ~ x2)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.62687 -0.75156 -0.03598  0.72383  2.44890 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   2.3899     0.1949   12.26  < 2e-16 ***
x2            2.8996     0.6330    4.58 1.37e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.072 on 98 degrees of freedom
Multiple R-squared:  0.1763,	Adjusted R-squared:  0.1679 
F-statistic: 20.98 on 1 and 98 DF,  p-value: 1.366e-05
```

All coefficients are significant. The estimate of β_0 is within 2 standard deviations of the true value. The estimate of β_1 is within 1 standard deviation of the true effective value.

f. No. Due to the collinearity, it is difficult to distinguish between the effect of x_1 and x_2 on y when they are regressed together. When regressed separately, the effect is much easier to see.

g. In all cases, the coefficient estimate for x_1 has decreased errantly, and that for x_2 has increased errantly. For the full model and for the x_2 only model, it is seen only has a high-leverage point. For the x_1 only model, it is seen as a high-leverage outlier.