

STK2100

Solutions to selected exercises in the book *An Introduction to Statistical Learning*

Vinnie Ko

April 4, 2018

Exercise 3.7.5

It's given that:

$$\hat{y}_i = x_i \hat{\beta} \quad \text{and} \quad \hat{\beta} = \frac{\sum_{k=1}^n x_k y_k}{\sum_{k'=1}^n x_{k'}^2}.$$

We then write:

$$\begin{aligned} \hat{y}_i &= x_i \hat{\beta} \\ &= x_i \frac{\sum_{k=1}^n x_k y_k}{\sum_{k'=1}^n x_{k'}^2} \\ &= \frac{\sum_{k=1}^n x_k y_k x_i}{\sum_{k'=1}^n x_{k'}^2} \\ &= \sum_{k=1}^n \left(\frac{x_i x_k}{\sum_{k'=1}^n x_{k'}^2} y_k \right) \\ &= \sum_{k=1}^n a_k y_k \quad \text{where} \quad a_k = \frac{x_i x_k}{\sum_{k'=1}^n x_{k'}^2}. \end{aligned}$$

So, the fitted values from linear regression ($\hat{y}_i$) can be written as linear combinations of the response values (y_i).

Exercise 3.7.6

$$\begin{aligned} \bar{y} &= \hat{\beta}_0 + \hat{\beta}_1 \bar{x} \\ &= (\bar{y} - \hat{\beta}_1 \bar{x}) + \hat{\beta}_1 \bar{x} \\ &= \bar{y} \end{aligned}$$

Thus, the least squares line always passes through the point $(\bar{x}, \bar{y})$.

Exercise 3.7.7

$$\begin{aligned}
\text{Cor}(\mathbf{y}, \mathbf{x})^2 &= \text{Cor}(\mathbf{y}, \mathbf{x}\hat{\beta})^2 \\
&= \text{Cor}(\mathbf{y}, \hat{\mathbf{y}})^2 \\
&= \left(\frac{\text{Cov}(\mathbf{y}, \hat{\mathbf{y}})}{\sqrt{\text{Var}(\mathbf{y})\text{Var}(\hat{\mathbf{y}})}} \right)^2 \\
&= \frac{\text{Cov}(\mathbf{y}, \hat{\mathbf{y}})^2}{\text{Var}(\mathbf{y})\text{Var}(\hat{\mathbf{y}})} \\
&= \frac{\text{Cov}(\hat{\mathbf{y}} + \boldsymbol{\varepsilon}, \hat{\mathbf{y}})^2}{\text{Var}(\mathbf{y})\text{Var}(\hat{\mathbf{y}})} \\
&= \frac{(\text{Cov}(\hat{\mathbf{y}}, \hat{\mathbf{y}}) + \text{Cov}(\hat{\mathbf{y}}, \boldsymbol{\varepsilon}))^2}{\text{Var}(\mathbf{y})\text{Var}(\hat{\mathbf{y}})} \\
&= \frac{\text{Var}(\hat{\mathbf{y}})^2}{\text{Var}(\mathbf{y})\text{Var}(\hat{\mathbf{y}})} \\
&= \frac{\text{Var}(\hat{\mathbf{y}})}{\text{Var}(\mathbf{y})} \\
&= \frac{\text{Var}(\hat{\mathbf{y}})}{TSS}
\end{aligned}$$

Now, we have to show

$$\text{Var}(\hat{\mathbf{y}}) = TSS - RSS = \text{Var}(\mathbf{y}) - \sum_{i=1}^n \varepsilon_i^2.$$

which is equal to showing

$$\sum_{i=1}^n (y_i - \bar{y})^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \sum_{i=1}^n (\hat{y}_i - \bar{y})^2.$$

We start from the left side.

$$\begin{aligned}
\sum_{i=1}^n (y_i - \bar{y})^2 &= \sum_{i=1}^n (y_i - \hat{y}_i + \hat{y}_i - \bar{y})^2 \\
&= \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \sum_{i=1}^n (\hat{y}_i - \bar{y})^2 + 2 \sum_{i=1}^n (y_i - \hat{y}_i)(\hat{y}_i - \bar{y})
\end{aligned}$$

We have to show that the last term is 0.

$$\begin{aligned}
2 \sum_{i=1}^n (y_i - \hat{y}_i)(\hat{y}_i - \bar{y}) &= 2 \sum_{i=1}^n (\hat{y}_i - \bar{y})\varepsilon_i \\
&= 2 \sum_{i=1}^n (\hat{\beta}_0 + x_i \hat{\beta}_1 - \bar{y})\varepsilon_i \\
&= 2\hat{\beta}_0 \sum_{i=1}^n \varepsilon_i + 2\hat{\beta}_1 \sum_{i=1}^n x_i \varepsilon_i - 2\bar{y} \sum_{i=1}^n \varepsilon_i \\
&= 0
\end{aligned}$$

This is because $\sum_{i=1}^n \varepsilon_i = 0$ and $\sum_{i=1}^n x_i \varepsilon_i = 0$.

$$\begin{aligned}
 (\text{Proof: } & \begin{bmatrix} \sum_{i=1}^n 1 \cdot \varepsilon_i \\ \sum_{i=1}^n x_{i,1} \varepsilon_i \\ \vdots \\ \sum_{i=1}^n x_{i,p} \varepsilon_i \end{bmatrix} = \mathbf{X}^T \boldsymbol{\varepsilon} = \mathbf{X}^T (\mathbf{y} - \hat{\mathbf{y}}) = \mathbf{X}^T (\mathbf{y} - \mathbf{X} \hat{\boldsymbol{\beta}}) = \mathbf{X}^T (\mathbf{y} - \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}) = \mathbf{X}^T \mathbf{y} - \mathbf{X}^T \mathbf{y} = 0)
 \end{aligned}$$

Finally, we have

$$\text{Cor}(\mathbf{y}, \mathbf{x})^2 = \frac{\text{Var}(\hat{\mathbf{y}})}{TSS} = \frac{TSS - RSS}{TSS} = R^2.$$

Exercise 3.7.8

```

> # Clean up the memory before we start.
> rm(list=ls(all=TRUE))
>
> # (a)
> # Load library
> library(ISLR)
>
> # Perform linear regression.
> lm.fit = lm(mpg ~ horsepower, data = Auto)
> # Display the summary of the model.
> 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

>
> # percentage RSE
> # Manually calculate RSS and RSE.
> RSS = sum((Auto[, "mpg"] - lm.fit$fitted.values)^2)

```

```

> RSE = sqrt(RSS/(392-2))
> RSE.percent = RSE/mean(Auto[, "mpg"])
> # Or do it in a semi-automatic way.
> RSE = summary(lm.fit)$sigma
> RSE.percent = RSE/mean(Auto[, "mpg"])
>
> # R-squared
> summary(lm.fit)$r.squared
[1] 0.6059483
> # Note that in the simple regression setting,  $R^2 = \text{cor}^2$ .
> cor(Auto[, "mpg"], Auto[, "horsepower"])^2
[1] 0.6059483
>
> # Predicted value of mpg when horsepower = 98.
>
> # Calculate confidence interval manually.
> as.numeric(predict(lm.fit, data.frame(horsepower=98)))
[1] 24.46708
> as.numeric(predict(lm.fit, data.frame(horsepower=98))) - qt(0.975, 392-2)*RSE*
sqrt(1/392 + (98 - mean(Auto[, "horsepower"]))^2/
sum((Auto[, "horsepower"] - mean(Auto[, "horsepower"]))^2))
[1] 23.97308
> as.numeric(predict(lm.fit, data.frame(horsepower=98))) + qt(0.975, 392-2)*RSE*
sqrt(1/392 + (98 - mean(Auto[, "horsepower"]))^2/
sum((Auto[, "horsepower"] - mean(Auto[, "horsepower"]))^2))
[1] 24.96108
>
> # Confidence interval
> predict(lm.fit, data.frame(horsepower=98), interval = "confidence", level = 0.95)
      fit      lwr      upr
1 24.46708 23.97308 24.96108
> # Prediction interval
> predict(lm.fit, data.frame(horsepower=98), interval = "predict", level = 0.95)
      fit      lwr      upr
1 24.46708 14.8094 34.12476
>
> # (b)
> # Plot the model with data points.
> plot(x = Auto[, "horsepower"], y = Auto[, "mpg"], xlab = "horsepower", ylab = "mpg")
> abline(lm.fit)
>
> # (c)
> # Display multiple plots on one page.
> par(mfrow=c(2,2))
>
> # Plot diagnostic plots of lm.fit
> plot(lm.fit)
>
> # Set the plot parameter back to default.
> par(mfrow=c(1,1))

```

(a)

i)

The p-value of β_1 is significant. So, we can conclude that there is a relationship between the predictor horsepower and the response mpg.

ii)

$$\text{RSE (residual standard error)} = \sqrt{\widehat{\text{Var}(\boldsymbol{\varepsilon})}} = \sqrt{\widehat{\sigma^2}} = \sqrt{\frac{1}{n-p} \sum_{i=1}^n \hat{\varepsilon}_i^2} = \sqrt{\frac{1}{n-p} \sum_{i=1}^n (y_i - \hat{y}_i)^2} = \sqrt{\frac{1}{n-p} \text{RSS}}$$

can be interpreted as the average amount that the response variable deviates from the regression line. According to the R output, $\text{RSE} = 4.906$.

The mean of response variable is 23.45. So, the percentage RSE is $20.92\% = \frac{4.906}{23.45}$.

$R^2 = 0.61$. This can be interpreted as a proportion of variance in Y that is explained by the model.

iii)

$\beta_1 = -0.16 < 0$. So, the predictor has a negative effect on the response.

iv)

Predicted value of y when $x = 98$:

$$\begin{aligned}\hat{y}_i &= \hat{\beta}_0 + \hat{\beta}_1 x_i \\ 24.47 &= 39.935861 - 0.157845 \cdot 98\end{aligned}$$

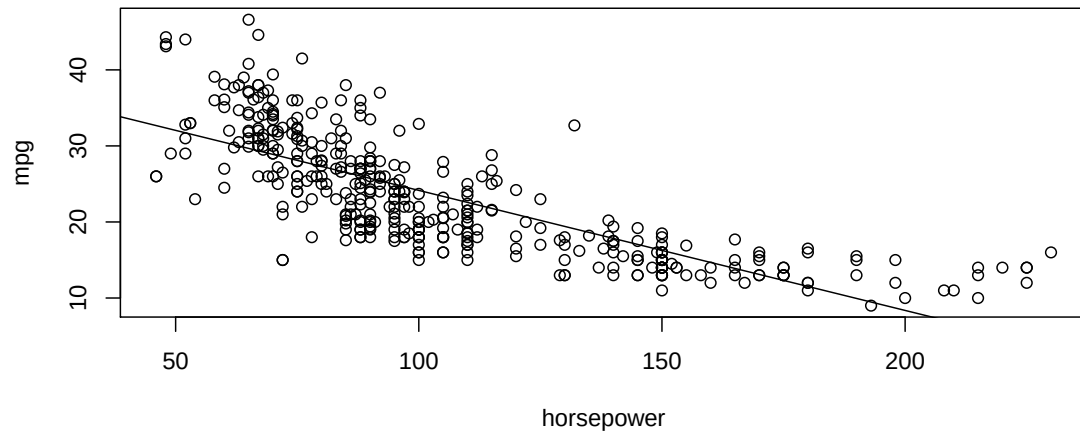
Confidence interval is an interval of $\widehat{E[Y|x]}$ (= sample mean, which is an estimator of population mean), which we obtained from the data and the corresponding fitted model. The significance level (e.g. 95%) indicates the probability that this interval captures $E[Y|x]$ (= population mean).

Prediction interval is an interval of $\hat{y}_{n+1}$, the prediction that we make with our estimated model and given $\mathbf{x}_{n+1}$. (Here we use the subscript $n+1$ to indicate that y_{n+1} is a value that is not yet observed. This is where the term *prediction* comes from.) The significance level (e.g. 95%) indicates the probability that this interval captures y_{n+1} .

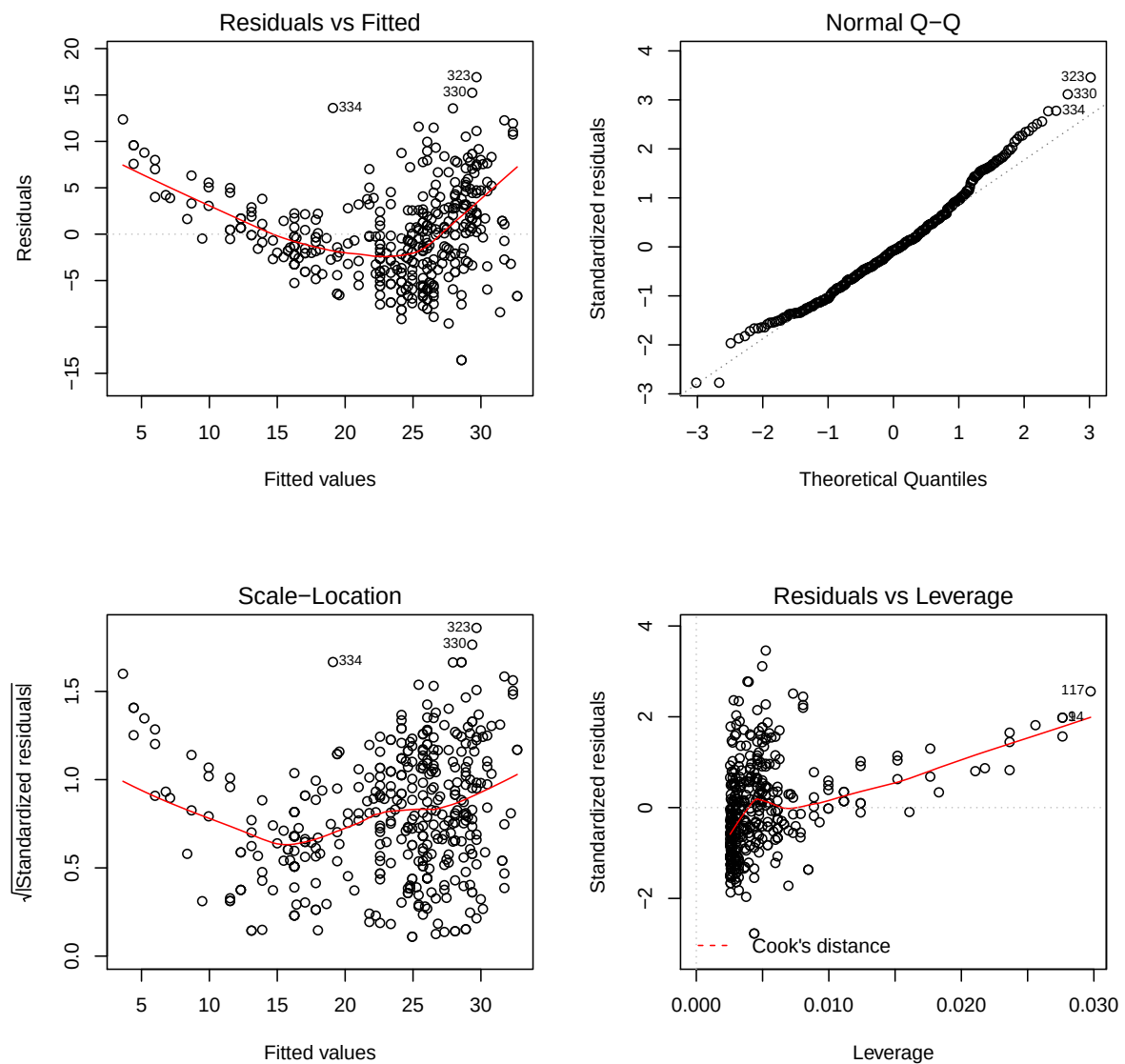
95% confidence interval from R is $[23.97, 24.96]$.

95% prediction interval from R is $[14.81, 34.12]$.

(b)



(c)



There is a visible trend in the residual plot. Which might indicate non-linearity of the data.

Exercise 3.7.9

```
> # Clean up the memory before we start.
> rm(list=ls(all=TRUE))
>
> # (a)
> # Load library
> library(ISLR)
>
> # Draw a scatterplot matrix
```

```

> pairs(Auto)
>
> # (b)
> # cor() only works between numerical variables. So, we remove non-numerical
  variables.
> numeric.columns = sapply(Auto, is.numeric)
>
> # Make a correlation matrix
> cor.matrix = cor(Auto[,numeric.columns])
>
> # (c)
> # Fit a multiple linear regression.
> lm.fit.1 = lm(mpg~.-name, data = Auto)
>
> # Display the result
> summary(lm.fit.1)

```

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

```

>
>
> # (d)
> # Display multiple plots on one page.
> par(mfrow=c(2,2))
>
> # Plot diagnostic plots of lm.fit
> plot(lm.fit.1)
>
> # Set the plot parameter back to default.
> par(mfrow=c(1,1))
>
>
> # (e)
> # Fit a multiple linear regression with interactions terms.

```

```

> lm.fit.2 = lm(mpg~.-name+cylinders*displacement+weight*displacement+horsepower*
  displacement, data = Auto)
>
>
> # (f)
> # Transform variables.
> # An example.
> lm.fit.3 = lm(log(mpg)~.-name, data = Auto)
> summary(lm.fit.3)

```

Call:

```
lm(formula = log(mpg) ~ . - name, data = Auto)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-0.40955	-0.06533	0.00079	0.06785	0.33925

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.751e+00	1.662e-01	10.533	< 2e-16 ***
cylinders	-2.795e-02	1.157e-02	-2.415	0.01619 *
displacement	6.362e-04	2.690e-04	2.365	0.01852 *
horsepower	-1.475e-03	4.935e-04	-2.989	0.00298 **
weight	-2.551e-04	2.334e-05	-10.931	< 2e-16 ***
acceleration	-1.348e-03	3.538e-03	-0.381	0.70339
year	2.958e-02	1.824e-03	16.211	< 2e-16 ***
origin	4.071e-02	9.955e-03	4.089	5.28e-05 ***

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

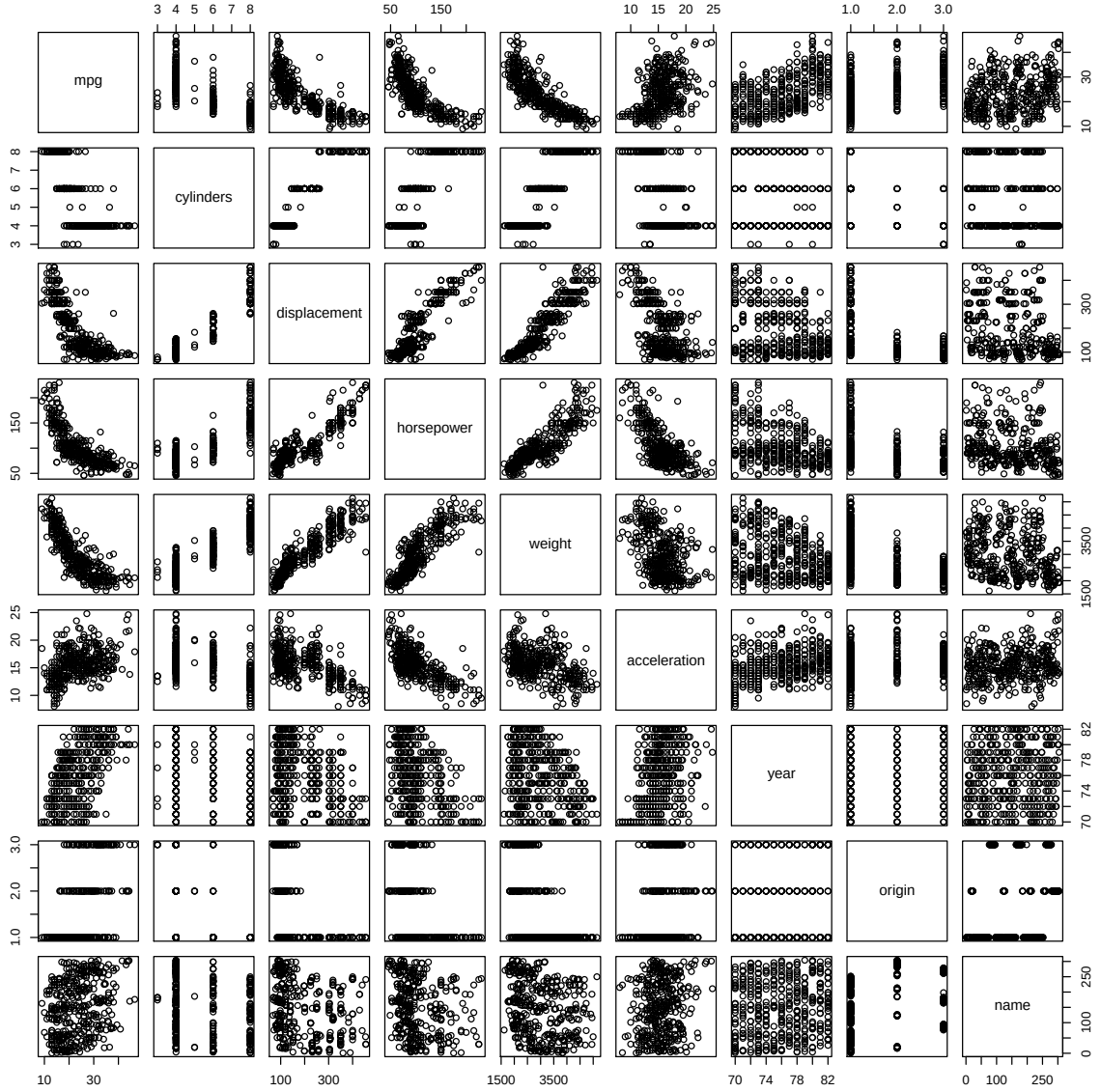
Residual standard error: 0.1191 on 384 degrees of freedom

Multiple R-squared: 0.8795, Adjusted R-squared: 0.8773

F-statistic: 400.4 on 7 and 384 DF, p-value: < 2.2e-16

```
> plot(lm.fit.3)
```

(a)



(b)

	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin
mpg	1.00	-0.78	-0.81	-0.78	-0.83	0.42	0.58	0.57
cylinders	-0.78	1.00	0.95	0.84	0.90	-0.50	-0.35	-0.57
displacement	-0.81	0.95	1.00	0.90	0.93	-0.54	-0.37	-0.61
horsepower	-0.78	0.84	0.90	1.00	0.86	-0.69	-0.42	-0.46
weight	-0.83	0.90	0.93	0.86	1.00	-0.42	-0.31	-0.59
acceleration	0.42	-0.50	-0.54	-0.69	-0.42	1.00	0.29	0.21
year	0.58	-0.35	-0.37	-0.42	-0.31	0.29	1.00	0.18
origin	0.57	-0.57	-0.61	-0.46	-0.59	0.21	0.18	1.00

Table 1: Correlation matrix of Auto.

(c)

i)

Yes. According to the F-test, there is a relationship between the model and the response.

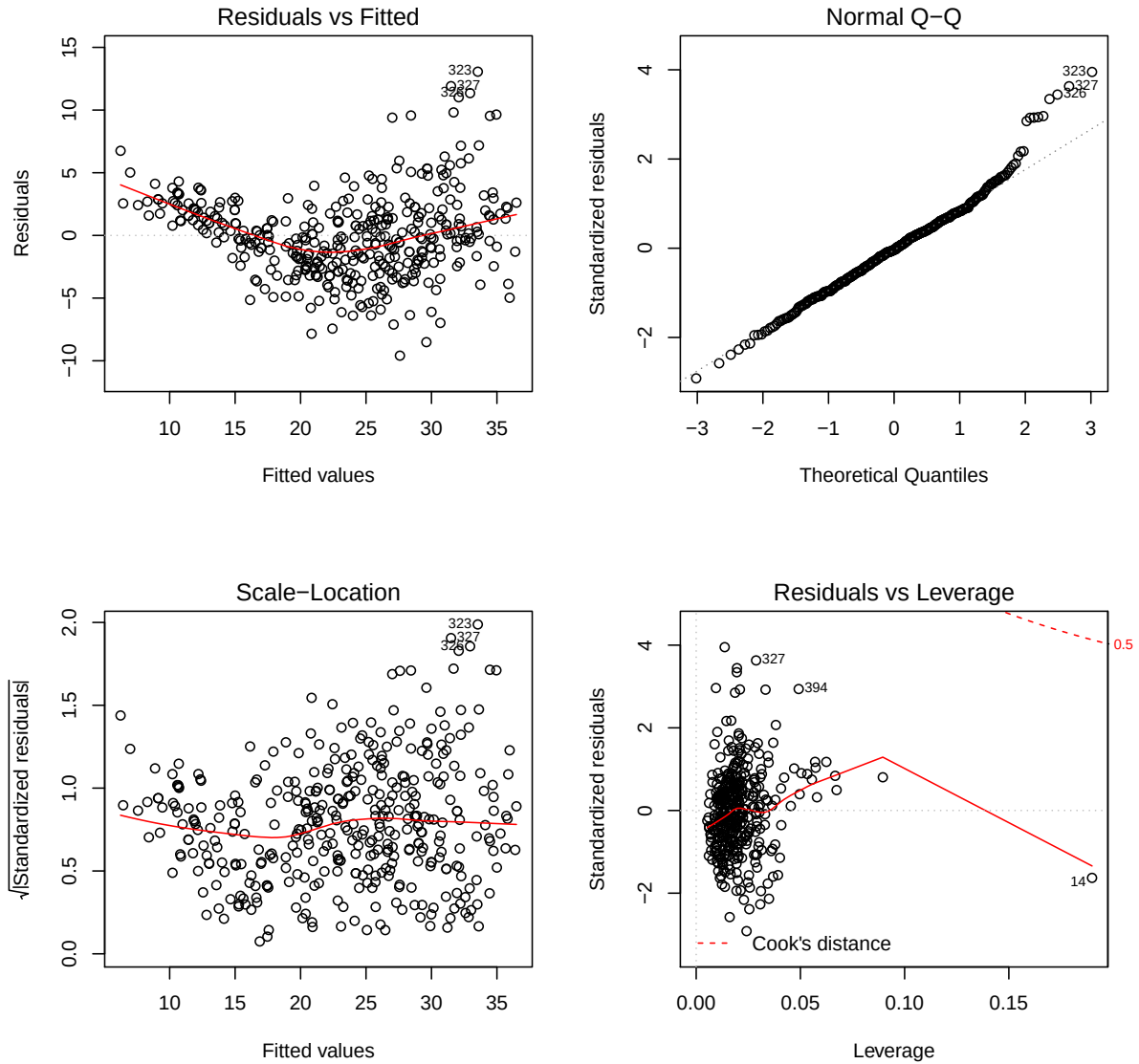
ii)

By looking at the p-values of $\hat{\beta}_j$, we can see that there is a statistically significant relationship between the response and `displacement`, `weight`, `year`, `origin`.

iii)

We see that $\hat{\beta}_{\text{year}} = 0.75$. If the value of all other predictors are fixed, increasing 1 in year will result in increase of 0.75 in mpg.

(d)



The visible trend in residual plot indicates that the fit is not completely optimal. From the leverage plot, we can see that point 14 has high leverage.

(e)

We look at the correlation matrix from (b) and include 3 highest correlated pairs into the linear model. The p-values of the fitted parameters ($\hat{\beta}_{\text{cylinders,displacement}}$, $\hat{\beta}_{\text{displacement,weight}}$, $\hat{\beta}_{\text{displacement,horsepower}}$) suggest that there 2 interaction effects: displacement:weight and displacement:horsepower.

(f)

There are many possibilities for transformations. Try some transformations on your own and look at the model fit and diagnostic plots to see what are the effects of the transformations.

Exercise 4.7.6

(a)

Logistic regression (GLM with logit link function):

$$\eta_i = \mathbf{x}_i \boldsymbol{\beta} \text{ and link function } g(\mu_i) = \ln \left(\frac{\mu_i}{1 - \mu_i} \right) = \eta_i.$$

The model:

$$p(X) = \frac{\exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2)}{1 + \exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2)} \quad \text{where } \beta_0 = -6, \beta_1 = 0.05, \beta_2 = 1$$

The probability that a student who studies for 40 hours and with an undergrad GPA of 3.5 gets an A:

$$0.3775 = \frac{\exp(-6 + 0.05 \cdot 40 + 3.5)}{1 + \exp(-6 + 0.05 \cdot 40 + 3.5)}.$$

(b)

By solving the following equation for X_1 :

$$0.5 = \frac{\exp(-6 + 0.05X_1 + 3.5)}{1 + \exp(-6 + 0.05X_1 + 3.5)}$$

we obtain that $X_1 = 50$.

So, the student from (a) need to study 50 hours to have a 50% change of getting an A.

Exercise 4.7.7

$$\text{Bayes theorem: } P(A | B) = \frac{P(B | A) P(A)}{P(B)}$$

In the classification setting, this theorem becomes

$$\Pr(Y = k | X = x) = \frac{\Pr^{\text{prior}}(Y = k) \Pr(X = x | Y = k)}{\sum_{l=1}^K \Pr^{\text{prior}}(Y = l) \Pr(X = x | Y = l)}, \quad \text{or in a simpler notation } p_k(x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^K \pi_l f_l(x)}$$

It's given in the exercise that X follows a normal distribution and

$$\{X | Y = 1\} \sim N(10, 36) \quad \text{and} \quad \{X | Y = 0\} \sim N(0, 36).$$

So,

$$\begin{aligned} p_{\text{yes}}(4) &= \frac{\pi_{\text{yes}} \exp(-\frac{1}{2\sigma^2}(x - \mu_{\text{yes}})^2)}{\pi_{\text{yes}} \exp(-\frac{1}{2\sigma^2}(x - \mu_{\text{yes}})^2) + \pi_{\text{no}} \exp(-\frac{1}{2\sigma^2}(x - \mu_{\text{no}})^2)} \\ &= \frac{0.8 \exp(-\frac{1}{2 \cdot 36}(4 - 10)^2)}{0.8 \exp(-\frac{1}{2 \cdot 36}(4 - 10)^2) + 0.2 \exp(-\frac{1}{2 \cdot 36}(4 - 0)^2)} \\ &= 0.7519 \end{aligned}$$

Exercise 4.7.9

(a)

$$\text{odds: } \frac{p}{1-p}$$

So, the information given in the exercise:

$$0.37 = \frac{p}{1-p}$$

By solving this equation, we obtain:

$$p = 0.27.$$

(b)

$$\text{odds: } \frac{0.16}{1-0.16} = 0.1905$$

Exercise 5.4.1

We have to show that

$$\frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}} = \arg \min_{\alpha} \text{Var}(\alpha X + (1-\alpha)Y).$$

Apply properties of (co)variance:

$$\begin{aligned} \text{Var}(\alpha X + (1-\alpha)Y) &= \text{Var}(\alpha X) + \text{Var}((1-\alpha)Y) + 2\text{Cov}(\alpha X, (1-\alpha)Y) \\ &= \alpha^2 \text{Var}(X) + (1-\alpha)^2 \text{Var}(Y) + 2\alpha(1-\alpha)\text{Cov}(X, Y) \\ &= \alpha^2 \sigma_X^2 + (\alpha-1)^2 \sigma_Y^2 - 2(\alpha^2 - \alpha)\sigma_{XY} \end{aligned}$$

Find the value of α that satisfies the first order condition (i.e. $\frac{\partial \text{Var}(\alpha X + (1-\alpha)Y)}{\partial \alpha} = 0$):

$$\begin{aligned} \frac{\partial \text{Var}(\alpha X + (1-\alpha)Y)}{\partial \alpha} &= \frac{\partial (\alpha^2 \sigma_X^2 + (\alpha-1)^2 \sigma_Y^2 - 2(\alpha^2 - \alpha)\sigma_{XY})}{\partial \alpha} \\ &= 2\alpha \sigma_X^2 + 2(\alpha-1)\sigma_Y^2 - 2(2\alpha-1)\sigma_{XY} \\ &= 0. \end{aligned}$$

By rewriting this in terms of α , we obtain:

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}}$$

Now, we check the second order condition:

$$\begin{aligned}
\frac{\partial^2 \text{Var}(\alpha X + (1 - \alpha)Y)}{\partial \alpha^2} &= \frac{\partial(2\alpha\sigma_X^2 + 2(\alpha - 1)\sigma_Y^2 - 2(2\alpha - 1)\sigma_{XY})}{\partial \alpha} \\
&= 2\sigma_X^2 + 2\sigma_Y^2 - 4\sigma_{XY} \\
&= 2\text{Var}(X - Y) \\
&\geq 0
\end{aligned}$$

Therefore given that $X \neq Y$,

$$\frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}} = \arg \min_{\alpha} \text{Var}(\alpha X + (1 - \alpha)Y).$$

Exercise 5.4.2

(a)

$$P(x_{\text{boot}} \neq x_j) = 1 - P(x_{\text{boot}} = x_j) = 1 - \frac{1}{n}$$

(b)

$$P(x_{\text{boot}} \neq x_j) = 1 - P(x_{\text{boot}} = x_j) = 1 - \frac{1}{n}$$

(c)

In this case, we sample independently n times from the original dataset *with replacement* (i.e. the same element can be drawn multiple times). So, the probability that the j th observation is not in the bootstrap sample (of size n) is:

$$\prod_{i=1}^n P(x_{\text{boot}} \neq x_j) = \prod_{i=1}^n (1 - P(x_{\text{boot}} = x_j)) = \left(1 - \frac{1}{n}\right)^n.$$

(d)

$$1 - \left(1 - \frac{1}{5}\right)^5 = 0.6723$$

(e)

$$1 - \left(1 - \frac{1}{100}\right)^{100} = 0.6340$$

(f)

$$1 - \left(1 - \frac{1}{10000}\right)^{10000} = 0.6321$$

(g)

$$1 - \left(1 - \frac{1}{100000}\right)^{100000} = 0.6321$$

```
> p.boot = function(n) {  
+   return(1 - (1 - 1/n)^n)  
+ }  
>  
> x = 1:10^5  
> y = p.boot(x)  
>  
> x2 = 1:20  
> y2 = p.boot(x2)  
>  
> # Display multiple plots on one page.  
> par(mfrow=c(2,1))  
> plot(x2, y2, xlab="n", ylab="p", font.main = 1, type="l")  
> plot(x, y, xlab="n", ylab="p", font.main = 1, type="l")
```

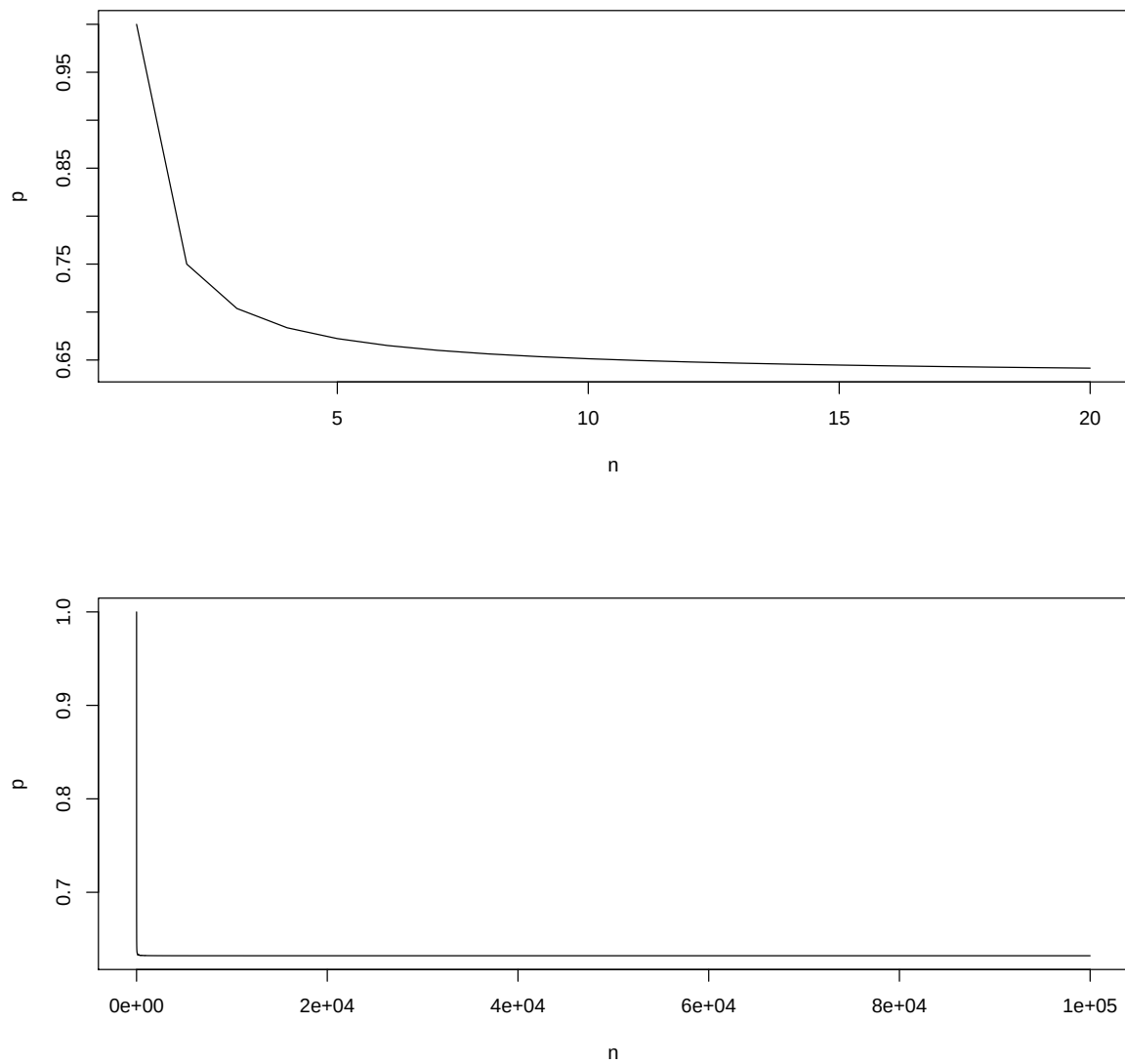


Figure 1: p : the probability that the j th observation is in the bootstrap sample.

(h)

```
> store = rep(NA, 10000)
> for (i in 1:10000) {
+   store[i] = sum(sample(1:100, rep=TRUE)==4)>0
+ }
> mean(store)
[1] 0.6302
```

Exercise 5.4.3

(a)

P.181 of the book explains the details of k -fold cross-validation.

(b)

i.

k -fold cross-validation is more robust to the randomness that is caused by which observations end up in training or test set.

k -fold cross-validation is computationally heavier.

ii.

k -fold cross-validation is computationally cheaper.

k -fold cross-validation still contains the randomness that is caused by which observations end up in training or test set.

k -fold cross-validation still contains the randomness that is caused by which observations end up in training or test set.

In general, k -fold cross-validation has higher bias than LOOCV since each training set contains relatively smaller sample size.

In general, k -fold cross-validation has smaller variance than LOOCV. Because in LOOCV, we are taking average of the outputs from highly correlated models.

Exercise 6.8.1

(a)

The model from *best subset selection* has the smallest training RSS because the other two methods search for the best model only within a part of all possible models.

(b)

In general, the model from *best subset selection*, will have the smallest test RSS because it considers all possible models as candidate, while the other two methods search for the best model within only a subset of all possible models. However, unlike in the training RSS situation, there is a chance that the model from *forward stepwise* or *backward stepwise* will outperform the model from *best subset selection* because of possible overfitting.

(c)

(i)

True

(ii)

True

(iii)
False

(iv)
False

(v)
False

Exercise 6.8.2

(a)

(iii) Lasso, relative to least squares, is less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

(b)

(iii) Ridge regression, relative to least squares, is less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

(c)

(ii) Non-linear methods, relative to least squares, are more flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

Exercise 6.8.3

(a)

(iv) Steadily decrease.

(b)

(ii) Decrease initially, and then eventually start increasing in a U shape.

(c)

(iii) Steadily increase. As s increases, the values of β s become more and more dependent on training data. So, the variance increases.

(d)

(iv) Steadily decrease.

(e)

(v) Remains constant.

Exercise 6.8.4

(a)

(iii) Steadily increase.

The higher λ , the more our $\hat{\beta}$ will deviate from $\hat{\beta}_{OLS}$ which is the minimizer of training RSS.

(b)

(ii) Decrease initially, and then eventually start increasing in a U shape.

When $\lambda = 0$, all β s have their least square estimate values. In this case, the model tries to fit hard to training data and hence test RSS can be high. As we increase λ , β s start reducing to 0 and some of the overfitting is reduced. Thus, test RSS initially decreases. Eventually, as β s approach 0, the model becomes too simple and test RSS increases.

(c)

(iv) Steadily decreases.

When $\lambda = 0$, β s have their least square estimate values. The actual estimates heavily depend on the training data and hence variance is high. As we increase λ , β s start decreasing and model becomes simpler. When $\lambda \rightarrow \infty$, all β s reduce to zero and the prediction from the model becomes a constant and has variance of 0.

(d)

(iii) Steadily increases.

When $\lambda = 0$, β s have their least-square estimate values and hence are unbiased. As λ increases, β s start reducing towards 0, the model fits less accurately to training data and hence bias increases. When $\lambda \rightarrow \infty$, the model predicts a constant and hence bias is maximum.

(e)

(v) Remains constant.

By definition, irreducible error is model independent and hence irrespective of the choice of λ , remains constant.

Exercise 6.8.5

(a)

The ridge regression optimization problem to find $\hat{\beta}$:

$$\hat{\beta} = \arg \min_{\beta} (\|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda\|\beta\|^2)$$

By using the given conditions, the optimization problem can be rewritten as:

$$\begin{aligned}\widehat{\boldsymbol{\beta}} &= \arg \min_{\boldsymbol{\beta}} \left((y_1 - \beta_1 x_{11} - \beta_2 x_{12})^2 + (y_2 - \beta_1 x_{21} - \beta_2 x_{22})^2 + \lambda(\beta_1^2 + \beta_2^2) \right) \\ &= \arg \min_{\boldsymbol{\beta}} \left(2(y_1 - x_{11}(\beta_1 + \beta_2))^2 + \lambda(\beta_1^2 + \beta_2^2) \right).\end{aligned}$$

(b)

To find the β_1 and β_2 that minimize this expression, we look at the first order condition.

$$\begin{aligned}\frac{\partial \text{RSS}_{\text{ridge}}}{\partial \beta_1} &= -4x_{11}(y_1 - x_{11}(\beta_1 + \beta_2)) + 2\lambda\beta_1 = 0 \\ \frac{\partial \text{RSS}_{\text{ridge}}}{\partial \beta_2} &= -4x_{11}(y_1 - x_{11}(\beta_1 + \beta_2)) + 2\lambda\beta_2 = 0\end{aligned}$$

which leads to

$$\widehat{\beta}_1 = \frac{2x_{11}(y_1 - x_{11}\beta_2)}{\lambda + 2x_{11}^2} \quad \text{and} \quad \widehat{\beta}_2 = \frac{2x_{11}(y_1 - x_{11}\beta_1)}{\lambda + 2x_{11}^2}.$$

By symmetry, $\widehat{\beta}_1 = \widehat{\beta}_2$.

(c)

The lasso optimization problem to find $\widehat{\boldsymbol{\beta}}$:

$$\widehat{\boldsymbol{\beta}} = \arg \min_{\boldsymbol{\beta}} \left(\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1 \right)$$

By using the given conditions, the optimization problem can be rewritten as:

$$\begin{aligned}\widehat{\boldsymbol{\beta}} &= \arg \min_{\boldsymbol{\beta}} \left((y_1 - \beta_1 x_{11} - \beta_2 x_{12})^2 + (y_2 - \beta_2 x_{21} - \beta_2 x_{22})^2 + \lambda(|\beta_1| + |\beta_2|) \right) \\ &= \arg \min_{\boldsymbol{\beta}} \left(2(y_1 - x_{11}(\beta_1 + \beta_2))^2 + \lambda(|\beta_1| + |\beta_2|) \right).\end{aligned}$$

(d)

Before we look at the first order condition, we first remove the absolute values by introducing c_1 and c_2 :

$$\begin{aligned}\widehat{\boldsymbol{\beta}} &= \arg \min_{\boldsymbol{\beta}} \left(2(y_1 - x_{11}(\beta_1 + \beta_2))^2 + \lambda(|\beta_1| + |\beta_2|) \right) \\ &= \arg \min_{\boldsymbol{\beta}} \left(2(y_1 - x_{11}(\beta_1 + \beta_2))^2 + \lambda(c_1\beta_1 + c_2\beta_2) \right)\end{aligned}$$

where

$$c_1 = \begin{cases} -1 & \text{if } \beta_1 < 0 \\ 1 & \text{if } \beta_1 \geq 0 \end{cases} \quad \text{and} \quad c_2 = \begin{cases} -1 & \text{if } \beta_2 < 0 \\ 1 & \text{if } \beta_2 \geq 0 \end{cases}.$$

We look at the first order condition.

$$\begin{aligned}\frac{\partial \text{RSS}_{\text{lasso}}}{\partial \beta_1} &= -4x_{11}(y_1 - x_{11}(\beta_1 + \beta_2)) + \lambda c_1 = 0 \\ \frac{\partial \text{RSS}_{\text{lasso}}}{\partial \beta_2} &= -4x_{11}(y_1 - x_{11}(\beta_1 + \beta_2)) + \lambda c_2 = 0\end{aligned}$$

which leads to

$$\hat{\beta}_1 = -\beta_2 + \frac{y_1}{x_{11}} - \frac{\lambda c_1}{4x_{11}^2} \quad \text{and} \quad \hat{\beta}_2 = -\beta_1 + \frac{y_1}{x_{11}} - \frac{\lambda c_2}{4x_{11}^2}.$$

So, there are multiple possible solutions depending on the sign of β_1 and β_2 .

Exercise 7.9.1

There are many ways to represent a cubic spline. Here, we write down a cubic spline as a cubic polynomial plus *truncated power basis* function ((7.10) from the book):

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x - \xi)_+^3$$

where

$$(x - \xi)_+^3 = \begin{cases} (x - \xi)^3 & \text{if } x > \xi \\ 0 & \text{otherwise} \end{cases}.$$

(a)

We want to represent a cubic spline with a cubic polynomial

$$f_1(x) = a_1 + b_1 x + c_1 x^2 + d_1 x^3$$

such that $f(x) = f_1(x)$ for all $x \leq \xi$.

When $x \leq \xi$, $(x - \xi)_+^3 = 0$. So, the cubic spline becomes

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$$

and we want $f_1(x)$ is equivalent to this.

If $a_1 = \beta_0, b_1 = \beta_1, c_1 = \beta_2, d_1 = \beta_3$, then $f(x) = f_1(x)$.

(b)

Now we look at the case when $x > \xi$. In this situation, a cubic spline is:

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x - \xi)^3$$

We want to represent a cubic spline with a cubic polynomial

$$f_2(x) = a_2 + b_2 x + c_2 x^2 + d_2 x^3$$

such that $f(x) = f_2(x)$ for all $x > \xi$.

We simply rewrite $f(x)$ so that it can be matched to $f_2(x)$:

$$\begin{aligned} f(x) &= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x - \xi)^3 \\ &= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x^3 - 3x^2 \xi + 3x \xi^2 - \xi^3) \\ &= (\beta_0 - \beta_4 \xi^3) + (\beta_1 + 3\beta_4 \xi^2)x + (\beta_2 - 3\beta_4 \xi)x^2 + (\beta_3 + \beta_4)x^3. \end{aligned}$$

If $a_2 = \beta_0 - \beta_4 \xi^3$, $b_2 = \beta_1 + 3\beta_4 \xi^2$, $c_2 = \beta_2 - 3\beta_4 \xi$, $d_2 = \beta_3 + \beta_4$, then $f(x) = f_2(x)$.

(c)

$$\begin{aligned} f_1(\xi) &= \beta_0 + \beta_1 \xi + \beta_2 \xi^2 + \beta_3 \xi^3 \\ f_2(\xi) &= (\beta_0 - \beta_4 \xi^3) + (\beta_1 + 3\beta_4 \xi^2)\xi + (\beta_2 - 3\beta_4 \xi)\xi^2 + (\beta_3 + \beta_4)\xi^3 \\ &= \beta_0 - \beta_4 \xi^3 + \beta_1 \xi + 3\beta_4 \xi^3 + \beta_2 \xi^2 - 3\beta_4 \xi^3 + \beta_3 \xi^3 + \beta_4 \xi^3 \\ &= \beta_0 + \beta_1 \xi + \beta_2 \xi^2 + 3\beta_4 \xi^3 - 3\beta_4 \xi^3 + \beta_3 \xi^3 + \beta_4 \xi^3 - \beta_4 \xi^3 \\ &= \beta_0 + \beta_1 \xi + \beta_2 \xi^2 + \beta_3 \xi^3 \end{aligned}$$

We have shown that $f_1(\xi) = f_2(\xi)$. Thus, $f(x)$ is continuous at ξ .

(d)

$$\begin{aligned} f'_1(x) &= \beta_1 + 2\beta_2 x + 3\beta_3 x^2 \\ f'_1(\xi) &= \beta_1 + 2\beta_2 \xi + 3\beta_3 \xi^2 \\ f'_2(x) &= (\beta_1 + 3\beta_4 \xi^2) + 2(\beta_2 - 3\beta_4 \xi)x + 3(\beta_3 + \beta_4)x^2 \\ f'_2(\xi) &= (\beta_1 + 3\beta_4 \xi^2) + 2(\beta_2 - 3\beta_4 \xi)\xi + 3(\beta_3 + \beta_4)\xi^2 \\ &= \beta_1 + 3\beta_4 \xi^2 + 2\beta_2 \xi - 6\beta_4 \xi^2 + 3\beta_3 \xi^2 + 3\beta_4 \xi^2 \\ &= \beta_1 + 2\beta_2 \xi + 3\beta_3 \xi^2 + 3\beta_4 \xi^2 + 3\beta_4 \xi^2 - 6\beta_4 \xi^2 \\ &= \beta_1 + 2\beta_2 \xi + 3\beta_3 \xi^2 \end{aligned}$$

We have shown that $f'_1(\xi) = f'_2(\xi)$. Thus, $f'(x)$ is continuous at ξ .

(e)

$$\begin{aligned} f''_1(x) &= 2\beta_2 + 6\beta_3 x \\ f''_1(\xi) &= 2\beta_2 + 6\beta_3 \xi \\ f''_2(x) &= 2(\beta_2 - 3\beta_4 \xi) + 6(\beta_3 + \beta_4)x \\ f''_2(\xi) &= 2(\beta_2 - 3\beta_4 \xi) + 6(\beta_3 + \beta_4)\xi \\ &= 2\beta_2 - 6\beta_4 \xi + 6\beta_3 \xi + 6\beta_4 \xi \\ &= 2\beta_2 + 6\beta_3 \xi \end{aligned}$$

We have shown that $f''_1(\xi) = f''_2(\xi)$. Thus, $f''(x)$ is continuous at ξ .

To sum up, we have shown that

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x - \xi)_+^3$$

satisfies all conditions of a cubic spline.

Thus, $f(x)$ is a proper representation of a cubic spline.

Exercise 7.9.2

Suppose that we use a smooth function $g(x)$ to fit a regression curve to the data (x, y) . And we compute $\hat{g}(x)$, the estimate of $g(x)$ by solving the following optimization problem:

$$\hat{g}(x) = \arg \min_g \left(\sum_{i=1}^n (y_i - g(x_i))^2 + \lambda \int [g^{(m)}(x)]^2 dx \right).$$

(a) $\lambda = \infty, m = 0$

Because $\lambda \rightarrow \infty$, the first term of the equation (RSS) will be ignored as long as the integral of the second term is not 0.

$\hat{g} = 0$, since it will minimize the second part of the equation.

(b) $\lambda = \infty, m = 1$

Because $\lambda \rightarrow \infty$, the first term of the equation (RSS) will be ignored as long as the integral of the second term is not 0.

$g = \beta_0$, since it will give $g'(x) = 0$ and the second term will be 0. $\hat{g} = \hat{\beta}_0^{\text{OLS}}$ will minimize the first term.

(c) $\lambda = \infty, m = 2$

Because $\lambda \rightarrow \infty$, the first term of the equation (RSS) will be ignored as long as the integral of the second term is not 0.

$g = \beta_0 + \beta_1 x$, since it will give $g''(x) = 0$ and the second term will be 0. $\hat{g} = \hat{\beta}_0^{\text{OLS}} + \hat{\beta}_1^{\text{OLS}} x$ will minimize the first term.

(d) $\lambda = \infty, m = 3$

Because $\lambda \rightarrow \infty$, the first term of the equation (RSS) will be ignored as long as the integral of the second term is not 0.

$g = \beta_0 + \beta_1 x + \beta_2 x^2$, since it will give $g'''(x) = 0$ and the second term will be 0. $\hat{g} = \hat{\beta}_0^{\text{OLS}} + \hat{\beta}_1^{\text{OLS}} x + \hat{\beta}_2^{\text{OLS}} x^2$ will minimize the first term.

(e) $\lambda = 0, m = 3$

Because $\lambda = 0$, the second term of the equation will be ignored.

The curve will be very wiggly to minimize RSS. In an extreme case, the curve will go through all data points and $\text{RSS} = 0$.

Exercise 7.9.4

$$y = \beta_0 + \beta_1 b_1(x) + \beta_2 b_2(x) + \varepsilon$$

where

$$b_1(x) = I(0 \leq x \leq 2) - (x-1)I(1 \leq x \leq 2) = \begin{cases} 1 & \text{if } 0 \leq x < 1 \\ -x+2 & \text{if } 1 \leq x \leq 2 \\ 0 & \text{otherwise} \end{cases}$$

$$b_2(x) = (x-3)I(3 \leq x \leq 4) + I(4 < x \leq 5) = \begin{cases} x-3 & \text{if } 3 \leq x < 4 \\ 1 & \text{if } 4 < x \leq 5 \\ 0 & \text{otherwise} \end{cases}.$$

The estimated curve is given by

$$\hat{y} = 1 + b_1(x) + 3b_2(x).$$

```
> x = seq(from = -2, to = 2, by = 0.01)
>
> b1 = Vectorize(
+   function(x) {
+     (x >= 0 && x <= 2) - (x - 1)*(x >= 1 && x <= 2)
+   }
+ )
>
> b2 = Vectorize(
+   function(x) {
+     (x - 3)*(x >= 3 && x <= 4) + (x > 4 && x <= 5)
+   }
+ )
>
> y = 1 + b1(x) + 3*(b2(x))
>
> plot(x, y, type = "l", ylab = quote(widehat("y")))
```

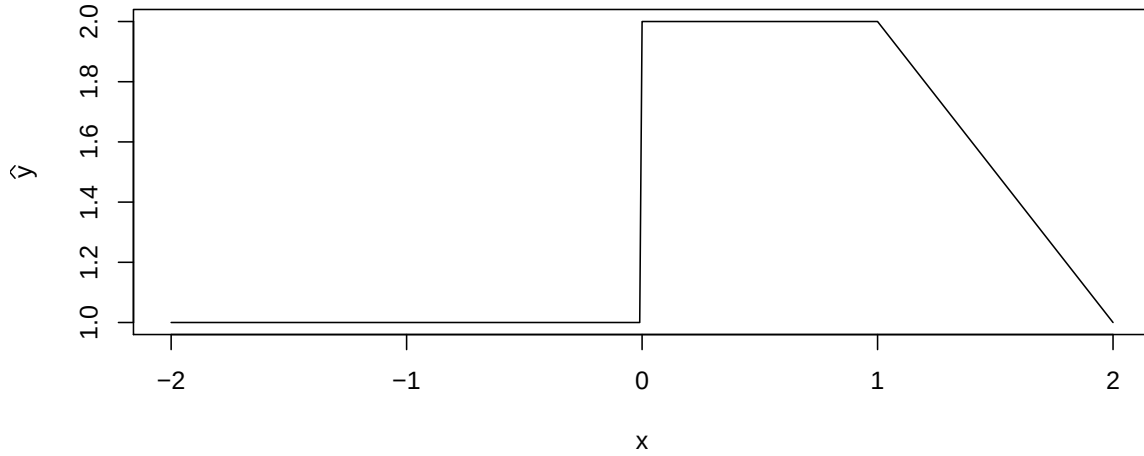


Figure 2: Result of exercise 7.9.4.

Exercise 7.9.5

(a)

We expect that $\hat{g}_2$ will have a smaller training RSS since $\hat{g}_2$ will tend to overfit.

(b)

We don't know. It depends on the complexity of true curve and data.

(c)

When $\lambda = 0$, the second term of the both equations will be ignored and the two optimizations problems will be the same. Thus, $\hat{g}_1 = \hat{g}_2$.

Exercise 8.4.2

Boosted model with B trees and p predictors:

$$\hat{g}(x) = \sum_{b=1}^B \lambda \hat{g}^b(x)$$

When we are using depth-one trees ($d = 1$), each tree within boosting contains only one split about only one predictor. So, we can rearrange $\hat{g}^b(x)$ with respect to the predictors:

$$\hat{g}(x) = \sum_{b=1}^B \lambda \hat{g}^b(x) = \sum_{i_1=1}^{K_1} \lambda \hat{g}_{i_1}(x_1) + \cdots + \sum_{i_j=1}^{K_j} \lambda \hat{g}_{i_j}(x_j) + \cdots + \sum_{i_p=1}^{K_p} \lambda \hat{g}_{i_p}(x_p)$$

where K_j indicates the number of trees that contains x_j and $\sum_{j=1}^p K_j = B$.

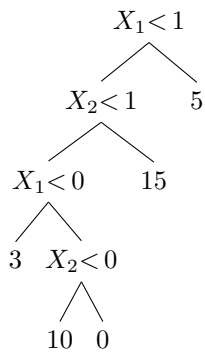
By defining $f_j(x_j) = \sum_{i_j=1}^{K_j} \lambda \hat{g}_{i_j}(x_j)$, we get

$$\begin{aligned}\hat{g}(x) &= \sum_{b=1}^B \lambda \hat{g}^b(x) \\ &= \sum_{i_1=1}^{K_1} \lambda \hat{g}_{i_1}(x_1) + \cdots + \sum_{i_j=1}^{K_j} \lambda \hat{g}_{i_j}(x_j) + \cdots + \sum_{i_p=1}^{K_p} \lambda \hat{g}_{i_p}(x_p) \\ &= \sum_{j=1}^p f_j(x_j)\end{aligned}$$

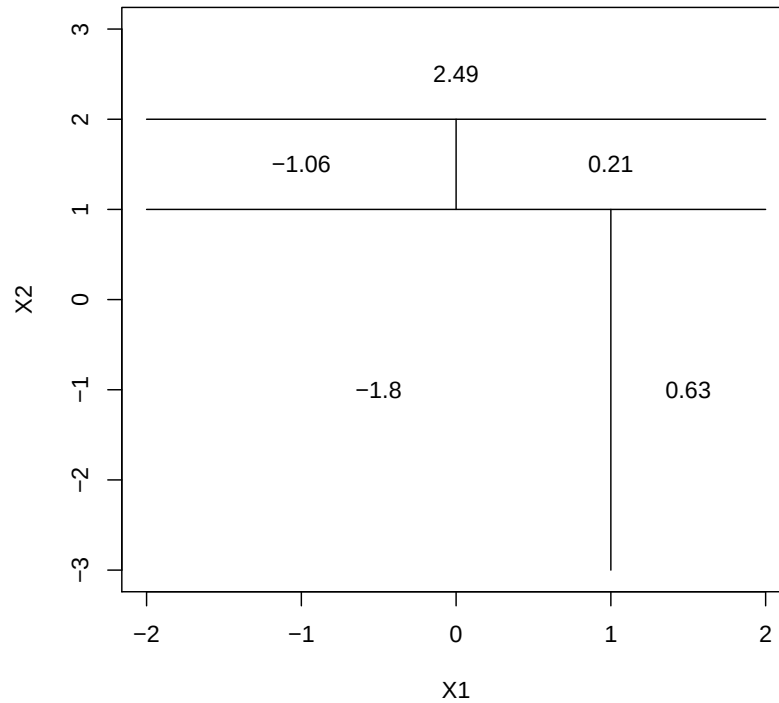
Thus, boosting using depth-one trees lead to an additive model.

Exercise 8.4.4

(a)



(b)



Exercise 8.4.5

i) Majority vote approach

$$P(\text{class in red} | x) = \begin{bmatrix} 0.1 \\ 0.15 \\ 0.2 \\ 0.2 \\ 0.55 \\ 0.6 \\ 0.6 \\ 0.65 \\ 0.7 \\ 0.75 \end{bmatrix} \quad \text{and this gives} \quad \hat{y} = \begin{bmatrix} \text{green} \\ \text{green} \\ \text{green} \\ \text{green} \\ \text{red} \\ \text{red} \\ \text{red} \\ \text{red} \\ \text{red} \\ \text{red} \end{bmatrix}$$

‘Red’ wins in voting. So, the final prediction is ‘red’.

ii) Average probability approach

$$\overline{P(\text{class in red} | x)} = \frac{0.1 + 0.15 + 0.2 + 0.2 + 0.55 + 0.6 + 0.6 + 0.65 + 0.7 + 0.75}{10} = 0.45$$

The probability of being 'green' is bigger. So, the final prediction is 'green'.