ISLR | Chapter 7 Exercises

Marshall McQuillen

9/21/2018

Conceptual

1

- **A**. The cubic piecewise polynomial:

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x-\xi)_+^3 \quad where \quad (x-\xi)_+^3 = \begin{cases} 0, & x \le \xi \\ (x-\xi)^3, & otherwise \end{cases}$$

  ...can be broken up and rewritten to be:

$$f(x) = \begin{cases} f_1(x) = a_1 + b_1 x + c_1 x^2 + d_1 x^3, & x \le \xi \\ f_2(x) = a_2 + b_2 x + c_2 x^2 + d_2 x^3, & otherwise \end{cases}$$

  In $f_1(x)$, since $(x-\xi)_+^3 = 0$ (because $x \le \xi$), the fifth term (of $f(x)$) zeroes out and the coefficients can be expresses as $a_1 = \beta_0,\ b_1 = \beta_1,\ c_1 = \beta_2$ and $d_1 = \beta_3$.

- **B**. Expanding the fifth term in $f(x)$ allows for the various powers of x to be grouped together and then recondensed. $a_2,\ b_2,\ c_2$ and d_2 are expressed in terms of the cofficients below.

$$\begin{aligned}
f_2(x) &= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x-\xi)^3 && (1)\\
&= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x-\xi)(x-\xi)(x-\xi) && (2)\\
&= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x^2 - 2x\xi + \xi^2)(x-\xi) && (3)\\
&= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x^3 - x^2\xi - 2x^2\xi + 2x\xi^2 + \xi^2 x - \xi^3) && (4)\\
&= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x^3 - 3x^2\xi + 3x\xi^2 - \xi^3) && (5)\\
&= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 x^3 - \beta_4 3x^2\xi + \beta_4 3x\xi^2 - \beta_4 \xi^3 && (6)\\
&= (\beta_0 - \beta_4\xi^3) + (\beta_1 x + \beta_4 3x\xi^2) + (\beta_2 x^2 - \beta_4 3x^2\xi) + (\beta_3 x^3 + \beta_4 x^3) && (7)\\
&= (\beta_0 - \beta_4\xi^3) + (\beta_1 + 3\beta_4\xi^2)x + (\beta_2 - 3\beta_4\xi)x^2 + (\beta_3 + \beta_4)x^3 && (8)\\
f_2(x) &= a_2 + b_2 x + c_2 x^2 + d_2 x^3 \quad where \quad \begin{cases} a_2 = \beta_0 - \beta_4\xi^3 \\ b_2 = \beta_1 + 3\beta_4\xi^2 \\ c_2 = \beta_2 - 3\beta_4\xi \\ d_2 = \beta_3 + \beta_4 \end{cases} && (9)
\end{aligned}$$

- **C**. Showing that $f(x)$ is continuous at ξ is illustrated by showing that $f(\xi)_1 = f(\xi)_2$.

$$
\begin{aligned}
f_1(\xi) &= a_1 + b_1(\xi) + c_1(\xi)^2 + d_1(\xi)^3 && (10)\\
&= \beta_0 + \beta_1(\xi) + \beta_2(\xi)^2 + \beta_3(\xi)^3 && (11)\\
& && (12)\\
f_2(\xi) &= a_2 + b_2(\xi) + c_2(\xi)^2 + d_2(\xi)^3 && (13)\\
&= (\beta_0 - \beta_4\xi^3) + (\beta_1 + 3\beta_4\xi^2)(\xi) + (\beta_2 - 3\beta_4\xi)(\xi)^2 + (\beta_3 + \beta_4)(\xi)^3 && (14)\\
&= (\beta_0 - \beta_4\xi^3) + (\beta_1\xi + 3\beta_4\xi^3) + (\beta_2\xi^2 - 3\beta_4\xi^3) + (\beta_3\xi^3 + \beta_4\xi^3) && (15)\\
&= \beta_0 - \beta_4\xi^3 + \beta_1\xi + 3\beta_4\xi^3 + \beta_2\xi^2 - 3\beta_4\xi^3 + \beta_3\xi^3 + \beta_4\xi^3 && (16)\\
&= \beta_0 + \beta_1\xi + \beta_2\xi^2 + \beta_3\xi^3 + 3\beta_4\xi^3 - 3\beta_4\xi^3 + \beta_4\xi^3 - \beta_4\xi^3 && (17)\\
&= \beta_0 + \beta_1\xi + \beta_2\xi^2 + \beta_3\xi^3 + (3\beta_4\xi^3 - 3\beta_4\xi^3) + (\beta_4\xi^3 - \beta_4\xi^3) && (18)\\
f_2(\xi) &= \beta_0 + \beta_1\xi + \beta_2\xi^2 + \beta_3\xi^3 && (19)
\end{aligned}
$$

$$f_2(\xi) = \beta_0 + \beta_1\xi + \beta_2\xi^2 + \beta_3\xi^3 = f_1(\xi)$$

- **D**. In order to show that $f_1'(\xi) = f_2'(\xi)$, we must first find $f'(x)$ with respect to x and then simplify both $f_1'(\xi)$ and $f_2'(\xi)$.

$$
\begin{aligned}
f(x) &= a_1 + b_1x + c_1x^2 + d_1x^3 && (20)\\
f'(x) &= b_1 + 2c_1x + 3d_1x^2 && (21)
\end{aligned}
$$

Therefore, substituting the necessary coefficients in for $b_1,\ c_1$ and d_1 in both $f_1'(\xi)$ and $f_2'(\xi)$, we get:

$$f'(x) = b_1 + 2c_1x + 3d_1x^2 \quad then \quad \begin{cases} f_1'(\xi) = \beta_1 + 2\beta_2\xi + 3\beta_3\xi^2 \\ f_2'(\xi) = (\beta_1 + 3\beta_4\xi^2) + 2(\beta_2 - 3\beta_4\xi)\xi + 3(\beta_3 + \beta_4)\xi^2 \end{cases} \tag{22}$$

$$
\begin{aligned}
f_2'(\xi) &= (\beta_1 + 3\beta_4\xi^2) + 2(\beta_2 - 3\beta_4\xi)\xi + 3(\beta_3 + \beta_4)\xi^2 && (23)\\
&= \beta_1 + 3\beta_4\xi^2 + 2\beta_2\xi - 6\beta_4\xi^2 + 3\beta_3\xi^2 + 3\beta_4\xi^2 && (24)\\
&= \beta_1 + 2\beta_2\xi + 3\beta_3\xi^2 + (3\beta_4\xi^2 + 3\beta_4\xi^2 - 6\beta_4\xi^2) && (25)\\
&= \beta_1 + 2\beta_2\xi + 3\beta_3\xi^2 + (6\beta_4\xi^2 - 6\beta_4\xi^2) && (26)\\
f_2'(\xi) &= \beta_1 + 2\beta_2\xi + 3\beta_3\xi^2 && (27)
\end{aligned}
$$

We now see that the derivative $f'(x)$ is continuous at knot ξ, which is to say $f_1'(\xi) = f_2'(\xi)$:

$$f_2'(\xi) = \beta_1 + 2\beta_2\xi + 3\beta_3\xi^2 = f_1'(\xi)$$

- **E**. In order to show that $f_1''(\xi) = f_2''(\xi)$, we must first find $f''(x)$ with respect to x and then simplify both $f_1''(\xi)$ and $f_2''(\xi)$.

$$\begin{aligned} f(x) &= a_1 + b_1 x + c_1 x^2 + d_1 x^3 && (28)\\ f'(x) &= b_1 + 2c_1 x + 3d_1 x^2 && (29)\\ f''(x) &= 2c_1 + 6d_1 x && (30) \end{aligned}$$

Therefore, substituting the necessary coefficients in for c_1 and d_1 in both $f_1''(\xi)$ and $f_2''(\xi)$, we come to:

$$f''(x) = 2c_1 + 6d_1 x \quad then \quad \begin{cases} f_1''(\xi) = 2\beta_2 + 6\beta_3\xi \\ f_2''(\xi) = 2(\beta_2 - 3\beta_4\xi) + 6(\beta_3 + \beta_4)\xi \end{cases} \tag{31}$$

$$\begin{aligned} f_2''(\xi) &= 2(\beta_2 - 3\beta_4\xi) + 6(\beta_3 + \beta_4)\xi && (32)\\ &= 2\beta_2 - 6\beta_4\xi + 6\beta_3\xi + 6\beta_4\xi && (33)\\ &= 2\beta_2 + 6\beta_3\xi + (6\beta_4\xi - 6\beta_4\xi) && (34)\\ f_2''(\xi) &= 2\beta_2 + 6\beta_3\xi && (35) \end{aligned}$$

We now see that the second derivative $f''(x)$ is continuous at knot ξ, which is to say $f_1''(\xi) = f_2''(\xi)$:

$$f_2''(\xi) = 2\beta_2 + 6\beta_3\xi = f_1''(\xi)$$

2

(sketches on following page)

- **A**. With $\lambda = \infty$, the second term will dominate the above equation and the RSS will be ignored. Since $g^0 = g$, this comes out to finding $g(x)$ that minimizes the integral of $g(x)$. Therefore, $g(x) = 0$.
- **B**. With $\lambda = \infty$ and $m = 1$, the second term will dominate the above equation and the RSS will be ignored. This then becomes a problem of finding a function $g(x)$ where $\int g'(x)$ is minimized. Therefore, $g(x) = c$ (a flat line) where c is a constant, ensuring that $g'(x) = 0$.
- **C**. With $\lambda = \infty$ and $m = 2$, the second term will dominate the above equation and the RSS will be ignored. This then becomes a problem of finding a function $g(x)$ where $\int g''(x)$ is minimized.

  If we work backwards conceptually, we will see that $g(x) = \beta_0 + \beta_1 x$. Since $\int g''(x)$ must be minimized, $g''(x) = 0$. Therefore, $g'(x) = c$ where c is some constant. This implies that $g(x)$ must have a constant slope, c aka β_1. Therefore, $g(x) = \beta_0 + \beta_1 x$
- **D**. With $\lambda = \infty$ and $m = 3$, the second term will dominate the above equation and the RSS will be ignored. This then becomes a problem of finding a function $g(x)$ where $\int g'''(x)$ is minimized. Therefore, $g(x) = \beta_0 + \beta_1 x + \beta_2 x^2$, $g(x)$ will be quadratic in some sense

  Once again, working backwards conceptually, if the goal is to minimize $\int g'''(x)$, then $g'''(x) = 0$. Therefore, $g''(x) = c$, where c is some constant. This implies that $g'(x)$ must have a constant slope, c. if $g'(x)$ has a constant slope, then $g(x) = \beta_0 + \beta_1 x + \beta_2 x^2$. Having a quadratic equation means that the slope of $g(x)$ is changing at a fixed rate, which satisfies our condition that $g'(x) = c$.
- **E**. With $\lambda = 0$ and $m = 3$, the second term in the equation is completely ignored, and $g(x)$ becomes the line that interpolates all data points.

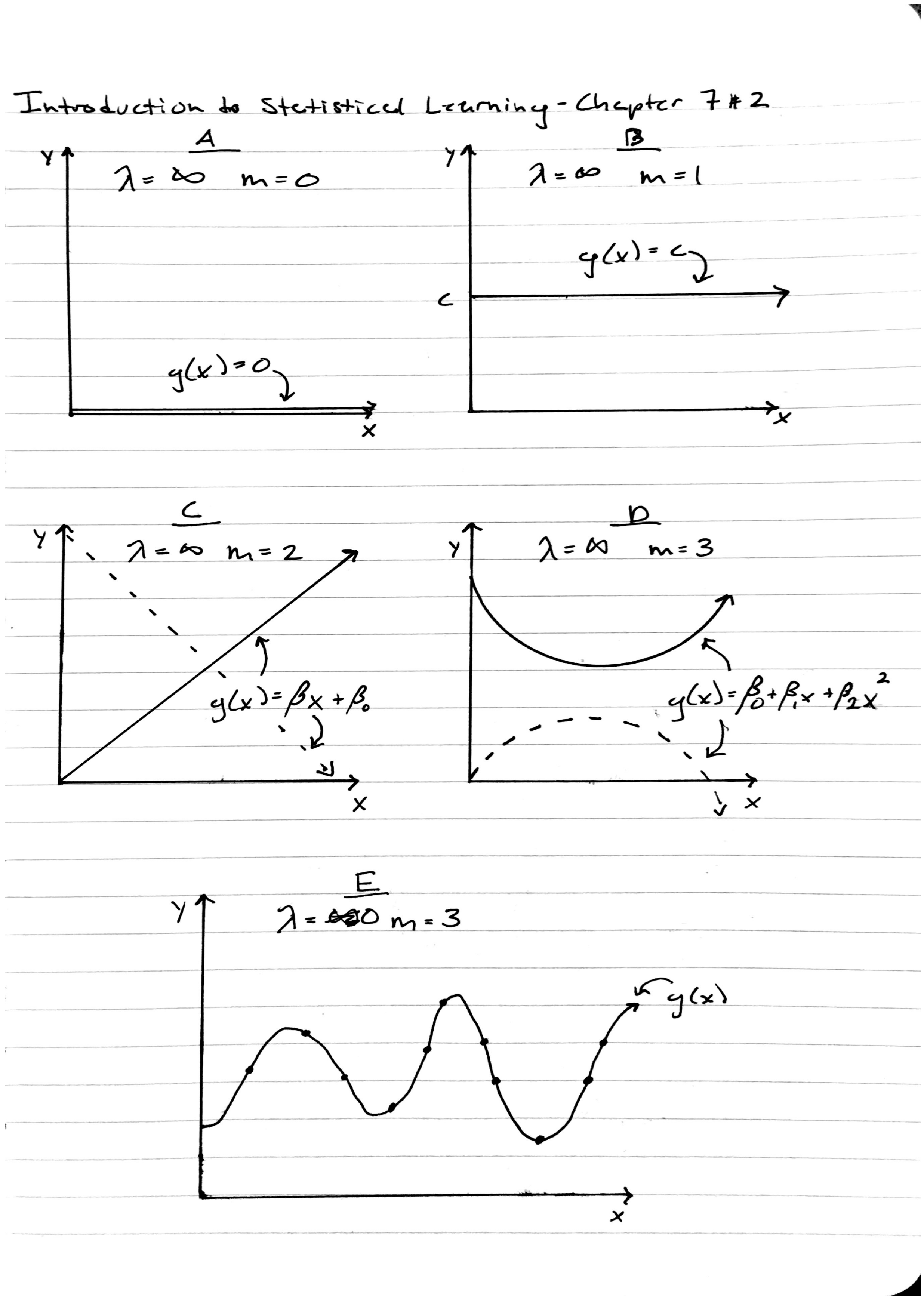
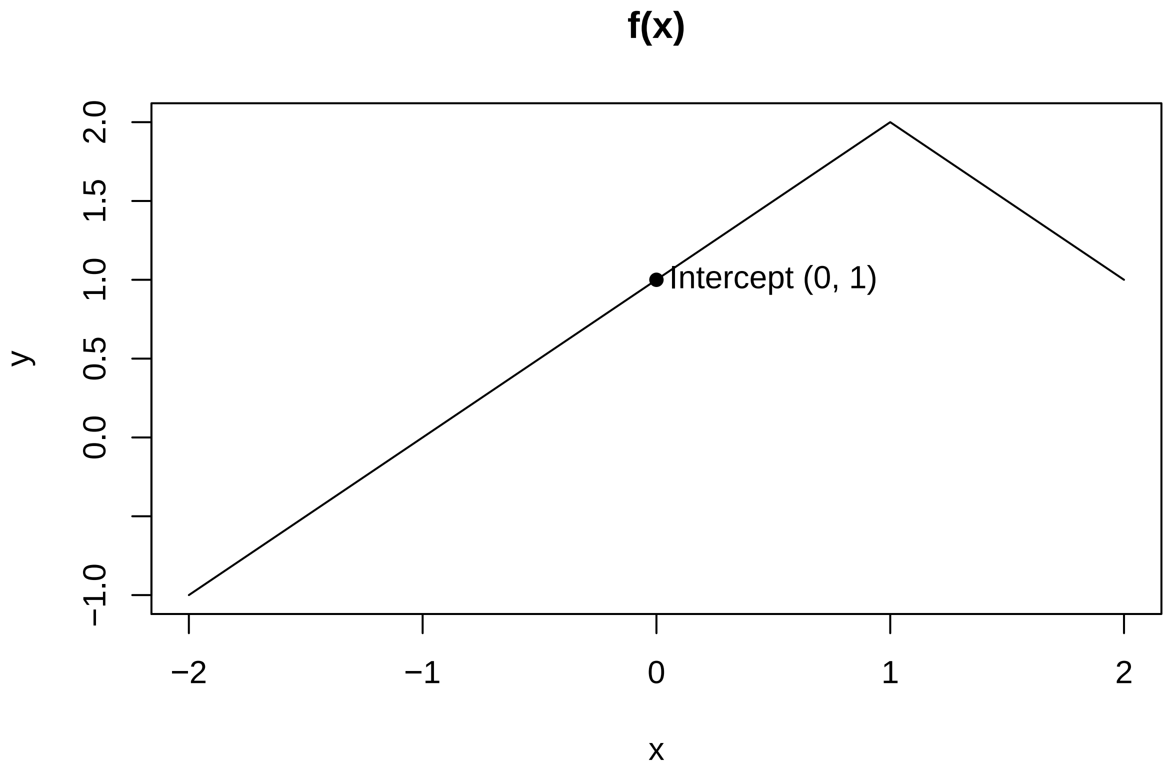


Figure 1: "Conceptual Exercise 2"

3

$$f(x) = 1 + x + \begin{cases} -2(x-1)^2, & x \geq 1 \\ 0, & otherwise \end{cases}$$

The intercept is at $y = 1$, $f(x)$ is linear with a slope equal to 1 up to $x = 1$, after which it becomes quadratic.

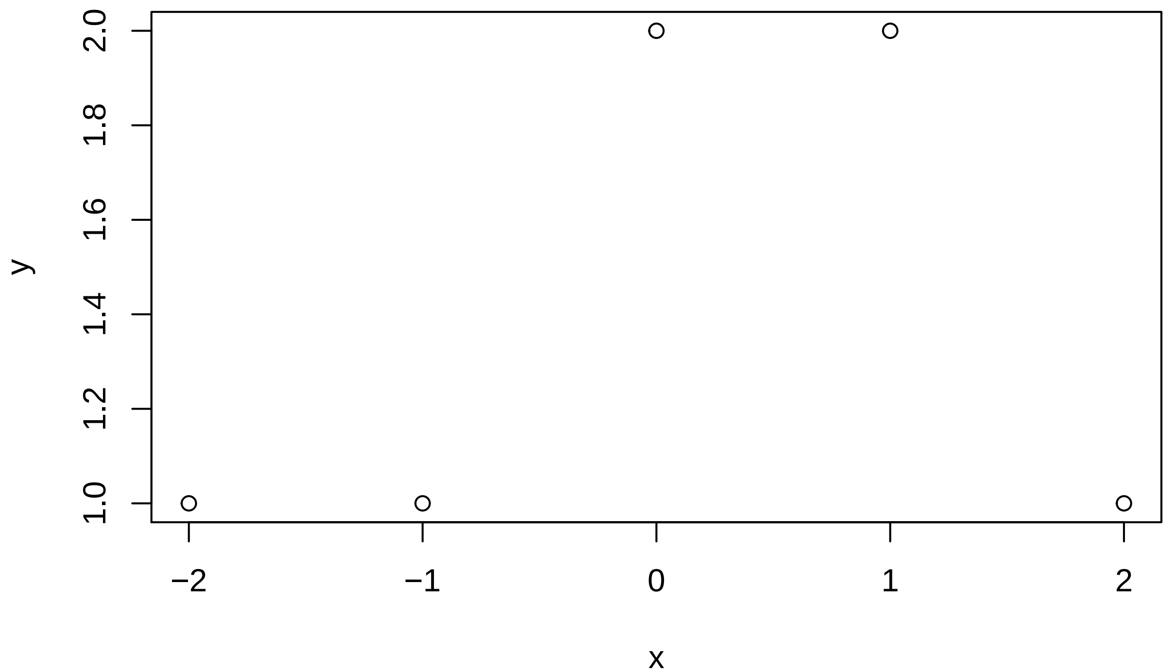


4

$$f(x) = \beta_0 + \beta_1 b_1(x) + \beta_2 b_2(x) \tag{36}$$

$$f(x) = 1 + b_1(x) + 3b_2(x) \quad where \quad \begin{cases} b_1(x) = I(0 \le x \le 2) - (x-1)I(1 \le x \le 2) \\ b_2(x) = (x-3)I(3 \le x \le 4) + I(4 < x \le 5) \end{cases} \tag{37}$$

```r
x <- -2:2
y <- c(1,1,2,2,1)
plot(x, y)
```



5

$$\hat{g_1} = \left(\sum_{i=1}^{n} (y_i - g(x_i))^2 + \int [g^3(x)]^2 dx \right) \tag{38}$$

$$\hat{g_2} = \left(\sum_{i=1}^{n} (y_i - g(x_i))^2 + \int [g^4(x)]^2 dx \right) \tag{39}$$

- **A**. As $\lambda \to \infty$, $\hat{g_2}$ will have a smaller training RSS. This is because $\hat{g_2}$ has one more degree of freedom than $\hat{g_1}$; in other words, it is allowed to be more flexible thatn $\hat{g_1}$.
- **B**. As $\lambda \to \infty$, $\hat{g_1}$ will most likely have a lower test RSS, although this is less certain than part **A**. It will most likely have a lower test RSS because we are constraining it more, which is to say there is less of a chance that it incorporates the error term ϵ into the model itself.
- **C**. If $\lambda = 0$, the two equations are the same so they will have the same training and test RSS (one that interpolates all data points).

Applied

6

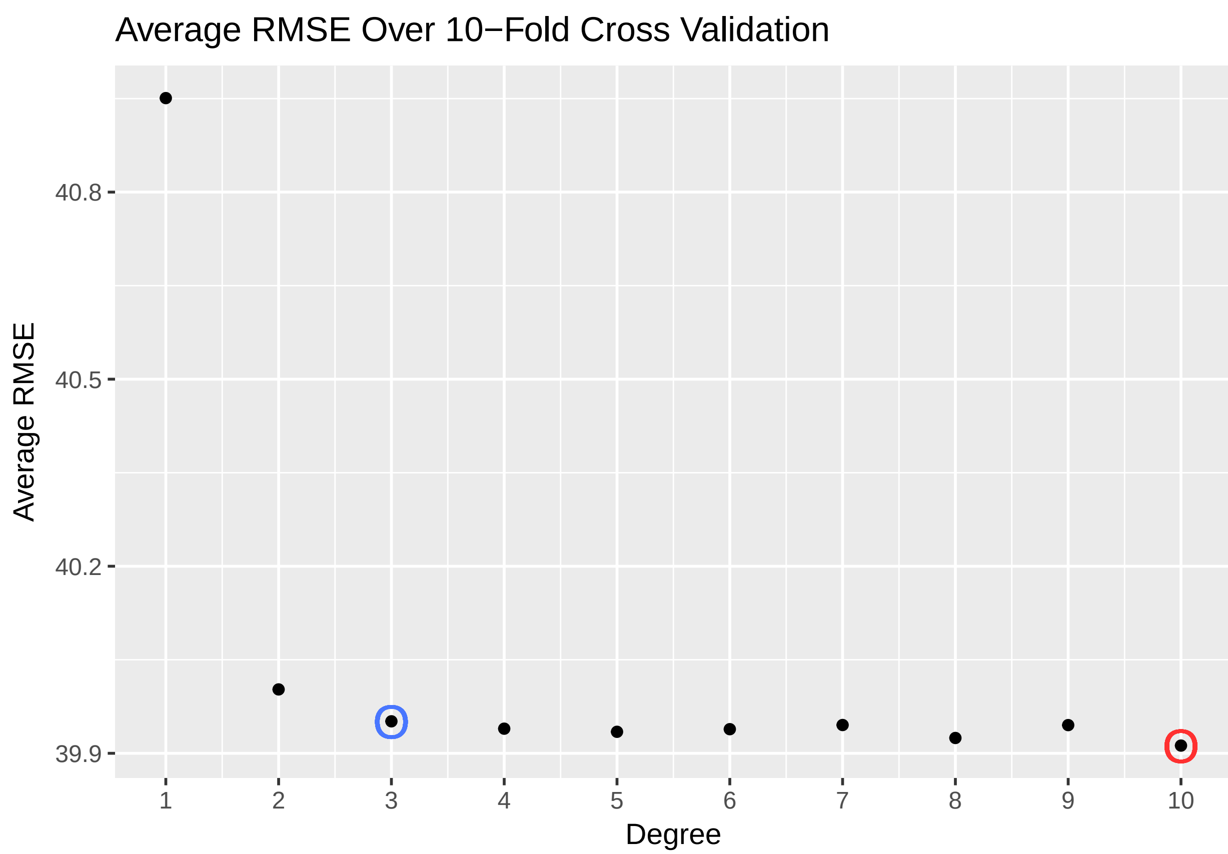
- **A**. Using 10-Fold CV of wage predicted by age for polynomial fits ranging in degree from 1 to 10, the minimus MSE is at a degree of 10. However, the RMSE only improves marginally after a third degree polynomial. Therefore, since a more complex model is only justifiable when accompanied by a significant decrease in the error rate, I will move forward with the third degree polynomial (which coincides with the results obtained from ANOVA.

```r
# imports
suppressPackageStartupMessages(library(ISLR))
suppressPackageStartupMessages(library(caret))
suppressPackageStartupMessages(library(boot))
suppressPackageStartupMessages(library(ggplot2))
attach(Wage)

set.seed(5)

# 10-Fold CV of Polynomial models with degree 1 - 10
degrees <- 1:10
cv.errors <- rep(0, 10)
for (i in degrees) {
    cv.fit <- glm(wage ~ poly(age, i), data = Wage)
    cv.errors[i] <- cv.glm(Wage, cv.fit, K = 10)$delta[1]
}

# Plot of CV errors
g <- ggplot(data.frame(x=1:10, y=sqrt(cv.errors)), aes(x, y)) +
    geom_point() +
    geom_point(aes(x=which.min(cv.errors),
                   y=sqrt(cv.errors[which.min(cv.errors)])),
               color = 'firebrick1',
               shape = "O",
               size = 6) +
     geom_point(aes(x=3,
               y=sqrt(cv.errors[3])),
           color = 'royalblue1',
           shape = "O",
           size = 6) +
    scale_x_continuous(breaks = 1:10,
                     labels = as.character(c(1:10))) +
    ggtitle("Average RMSE Over 10-Fold Cross Validation") +
    xlab("Degree") +
    ylab("Average RMSE")
g
```

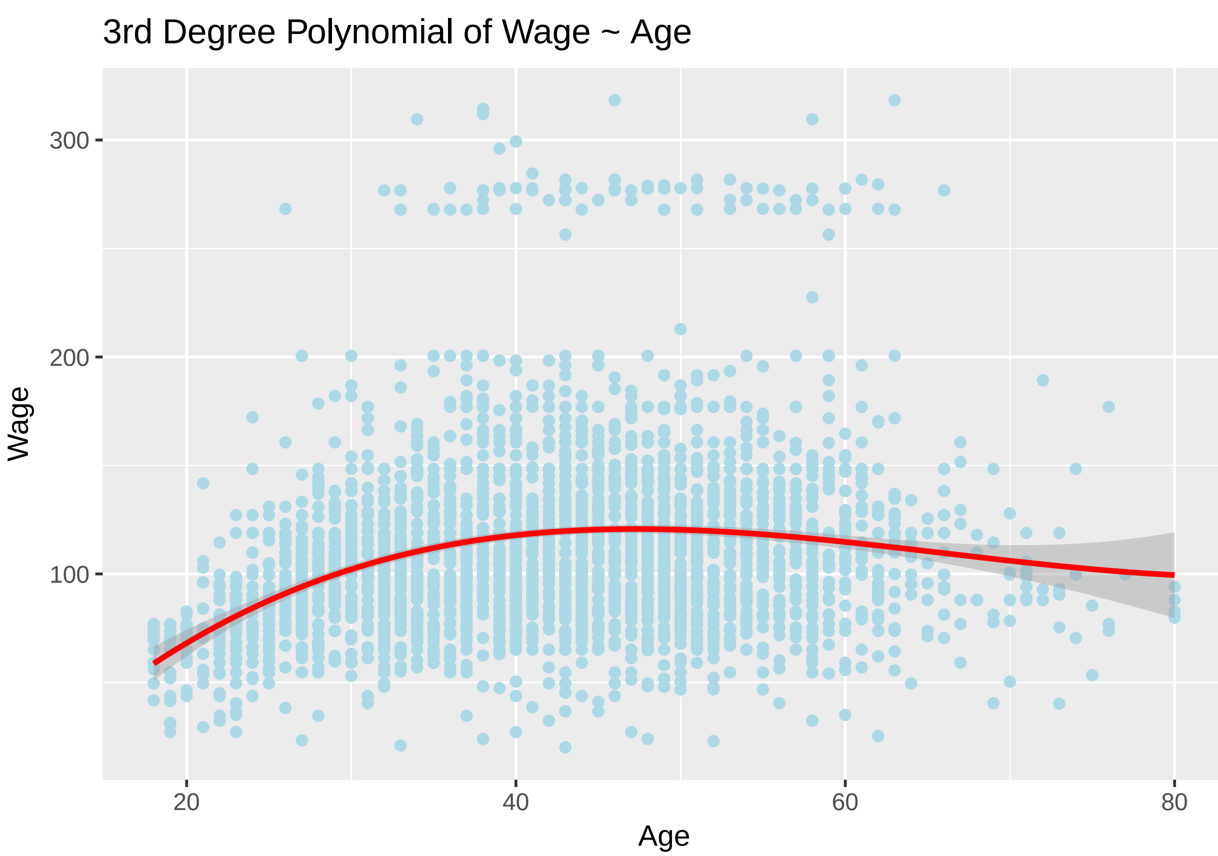



```r
# ANOVA
fit.1 <- lm(wage ~ age, data = Wage)
fit.2 <- lm(wage ~ poly(age, 2), data = Wage)
fit.3 <- lm(wage ~ poly(age, 3), data = Wage)
fit.4 <- lm(wage ~ poly(age, 4), data = Wage)
fit.5 <- lm(wage ~ poly(age, 5), data = Wage)
fit.6 <- lm(wage ~ poly(age, 6), data = Wage)
fit.7 <- lm(wage ~ poly(age, 7), data = Wage)
fit.8 <- lm(wage ~ poly(age, 8), data = Wage)
fit.9 <- lm(wage ~ poly(age, 9), data = Wage)
fit.10 <- lm(wage ~ poly(age, 10), data = Wage)
anova(fit.1, fit.2, fit.3, fit.4, fit.5, fit.6, fit.7, fit.8, fit.9, fit.10)
```

```
## Analysis of Variance Table
##
## Model  1: wage ~ age
## Model  2: wage ~ poly(age, 2)
## Model  3: wage ~ poly(age, 3)
## Model  4: wage ~ poly(age, 4)
## Model  5: wage ~ poly(age, 5)
## Model  6: wage ~ poly(age, 6)
## Model  7: wage ~ poly(age, 7)
## Model  8: wage ~ poly(age, 8)
## Model  9: wage ~ poly(age, 9)
## Model 10: wage ~ poly(age, 10)
##    Res.Df     RSS Df Sum of Sq       F    Pr(>F)
## 1    2998 5022216
```

```
## 2    2997 4793430  1    228786 143.7638 < 2.2e-16 ***
## 3    2996 4777674  1     15756   9.9005  0.001669 **
## 4    2995 4771604  1      6070   3.8143  0.050909 .
## 5    2994 4770322  1      1283   0.8059  0.369398
## 6    2993 4766389  1      3932   2.4709  0.116074
## 7    2992 4763834  1      2555   1.6057  0.205199
## 8    2991 4763707  1       127   0.0796  0.777865
## 9    2990 4756703  1      7004   4.4014  0.035994 *
## 10   2989 4756701  1         3   0.0017  0.967529
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Plot 3rd degree polynomial
g <- ggplot(Wage,
           aes(x = age, y = wage)) +
     geom_point(color = 'lightblue') +
     stat_smooth(method = 'lm',
                 formula = y ~ poly(x, 3),
                 size = 1,
                 color = 'red') +
     ggtitle("3rd Degree Polynomial of Wage ~ Age") +
     xlab("Age") +
     ylab("Wage")
g
```

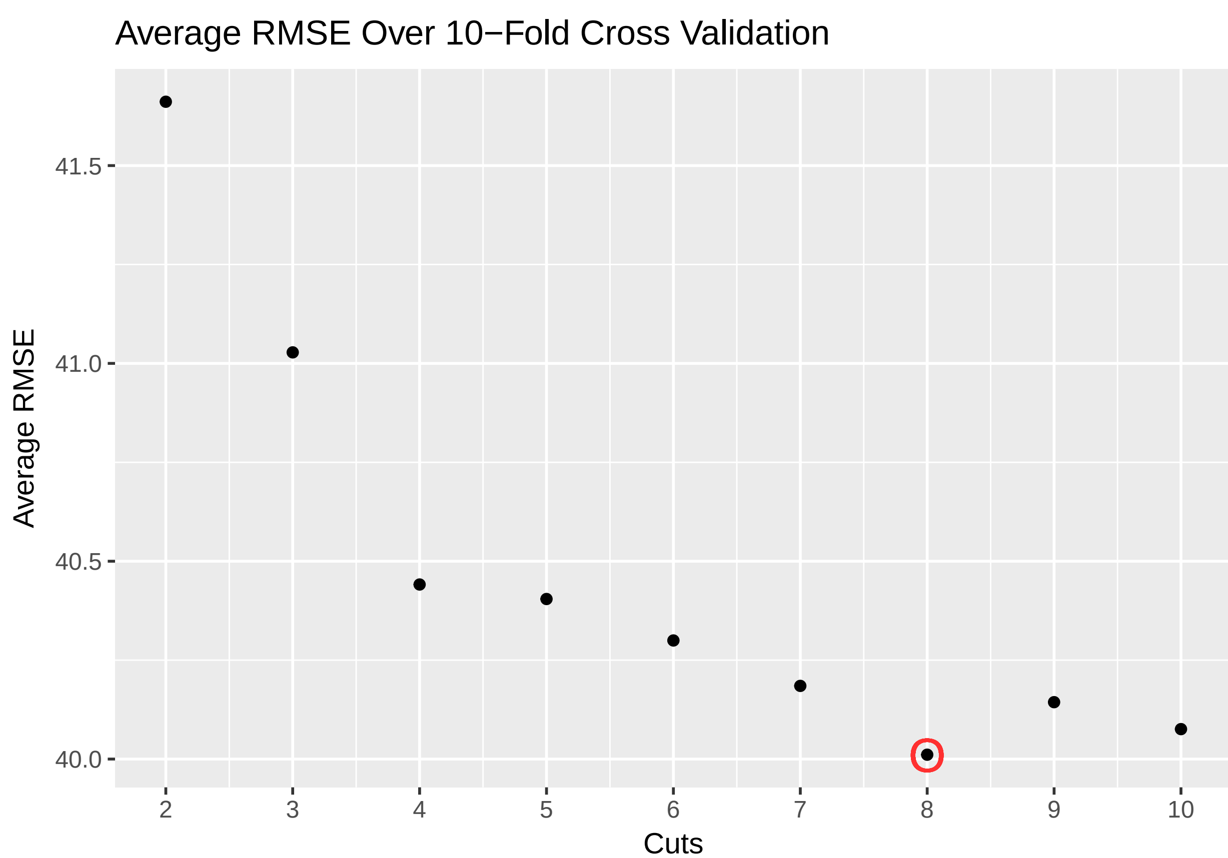


- **B**. Since the model will start to overfit as the number of cuts increases, I will limit the number of cuts to be a maximum of 10. As shown below, the minimum error is produced with 8 cuts.

```r
# 10-Fold CV of step functions up to 10 cuts
set.seed(5)

cuts <- 2:10
cv.errors <- rep(0, 9)
for (i in cuts) {
    Wage$age.cut <- cut(age, i)
    cv.fit <- glm(wage ~ age.cut, data = Wage)
    cv.errors[i-1] <- cv.glm(Wage, cv.fit, K = 10)$delta[1]
}

# Plot of CV error
g <- ggplot(data.frame(x=cuts, y=sqrt(cv.errors)), aes(x, y)) +
    geom_point() +
    geom_point(aes(x=which.min(cv.errors) + 1,
                   y=sqrt(cv.errors[which.min(cv.errors)])),
               color = 'firebrick1',
               shape = "O",
               size = 6) +
    scale_x_continuous(breaks = 1:10,
                       labels = as.character(c(1:10))) +
    ggtitle("Average RMSE Over 10-Fold Cross Validation") +
    xlab("Cuts") +
    ylab("Average RMSE")
g
```



```r
# Plot step function
g <- ggplot(Wage,
            aes(x = age, y = wage)) +
     geom_point(color = 'lightblue') +
     stat_smooth(method = 'lm',
                 formula = y ~ cut(x, 8),
                 size = 1,
                 color = 'red') +
     ggtitle("Stepwise Fit with 8 Cuts in Age Range") +
     xlab("Age") +
     ylab("Wage")
g
```