HW4

Peiran Chen

5/13/2022

1

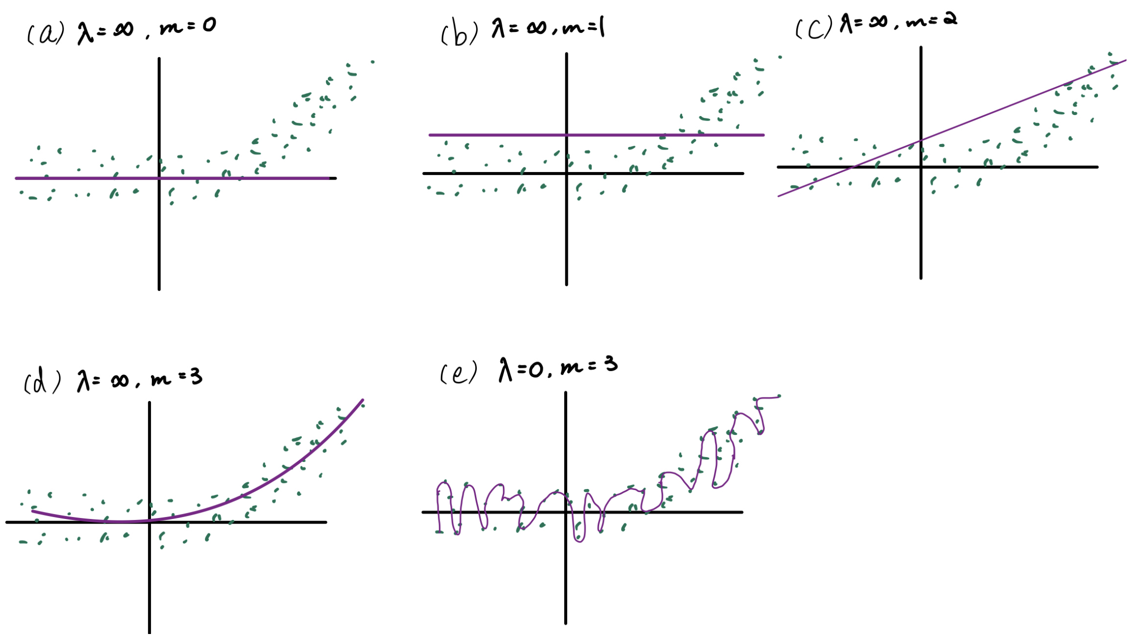


Figure 1: Question 1

For (a) to (d), when $\lambda = \infty$, we have a large penalty term that forces our function g to be smooth.

(a)

When $\lambda = \infty, m = 0$, $g^{(0)}(x) \to 0$. So, $\hat{g} = 0$.

(b)

When $\lambda = \infty, m = 1$, $g' = 0$. So, $\hat{g} = constant$, it would be a horizontal line.

(c)

When $\lambda = \infty, m = 2$, $g'' = 0$.So, $\hat{g}$ would be a straight line with slope.

(d)

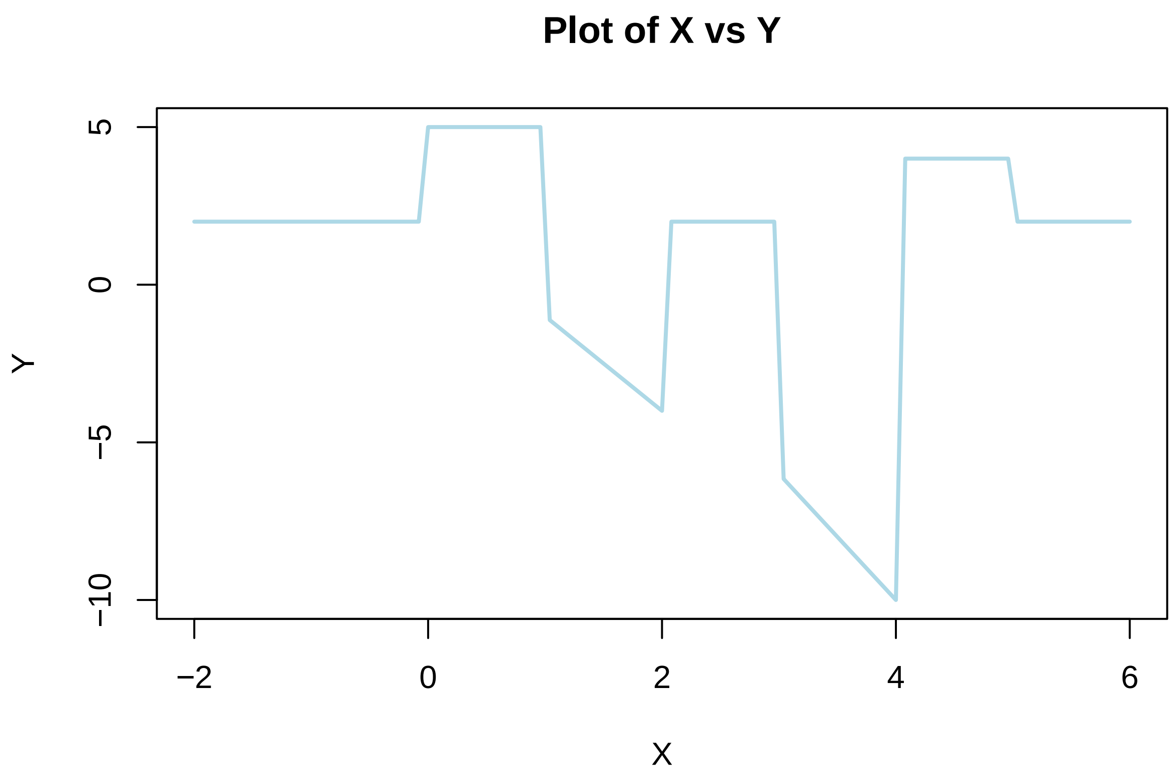
When $\lambda = \infty, m = 3$, $g''' = 0$. So, $\hat{g}$ woule be a curve that is quadratic.

(e)

When $\lambda = 0$ the penalty term has no effect in our function, so we get a curve/spline that interpolates all the n points perfectly.

2.

```
curve(2 + 3*(ifelse(x >= 0,
                   ifelse(x <= 2, 1, 0), 0) - (x+1)*ifelse(x >= 1,
                   ifelse(x <= 2, 1, 0), 0)) - 2 * ((2*x-2) * I(x>=3 & x <=4) -I(x > 4 & x <= 5)) ,
      from = -2,
      to   = 6,
      xlab = "X",
      ylab = "Y",
      col = "lightblue",
      lwd = 2,
      main = "Plot of X vs Y")
```



3.

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

- We can first find a cubic polynomial for $X \leqslant \psi$

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

- And we now have $f_1(X) = f(x)$. Then, for $X \geqslant \psi$.

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

- We now have shown that $f_2(X) = f(X)$. Then, we just need to show that $f(X)$ is continuous at ψ.
- At $X = \psi$:

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

Hence, we have

$$f_1(\psi) = f_2(\psi)$$

Therefore $f(X)$ is continuous at ψ.

- We now want to show that $f'(\psi) = f'(\psi)$ so that we can show $f'(X)$ is continuous at ψ.

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

Hence, we have

$$f_1'(\psi) = f_2'(\psi)$$

Therefore $f'(X)$ is continuous at ψ.

- We now want to show that $f''(\psi) = f''(\psi)$ so that we can show $f''(X)$ is continuous at ψ.

$$
\begin{aligned}
f_1''(\psi) &= 2\beta_2 + 6\beta_3\psi \\
f_2''(\psi) &= \frac{\partial}{\partial\psi} f_2'(\psi) \\
&= 2\beta_2 + 6\beta_3\psi
\end{aligned}
$$

Hence, we have

$$f_1''(\psi) = f_2''(\psi)$$

Therefore $f''(X)$ is continuous at ψ. Therefore, $f(x)$ is indeed a cubic spline.

4.

- To begin with, I first split the data into Training set and Validation set

```r
data(Wage)

set.seed(435)
# Train-Test Split
wage_split <- initial_split(Wage, strata = age)

wage_train <- training(wage_split)
wage_val  <- testing(wage_split)
```

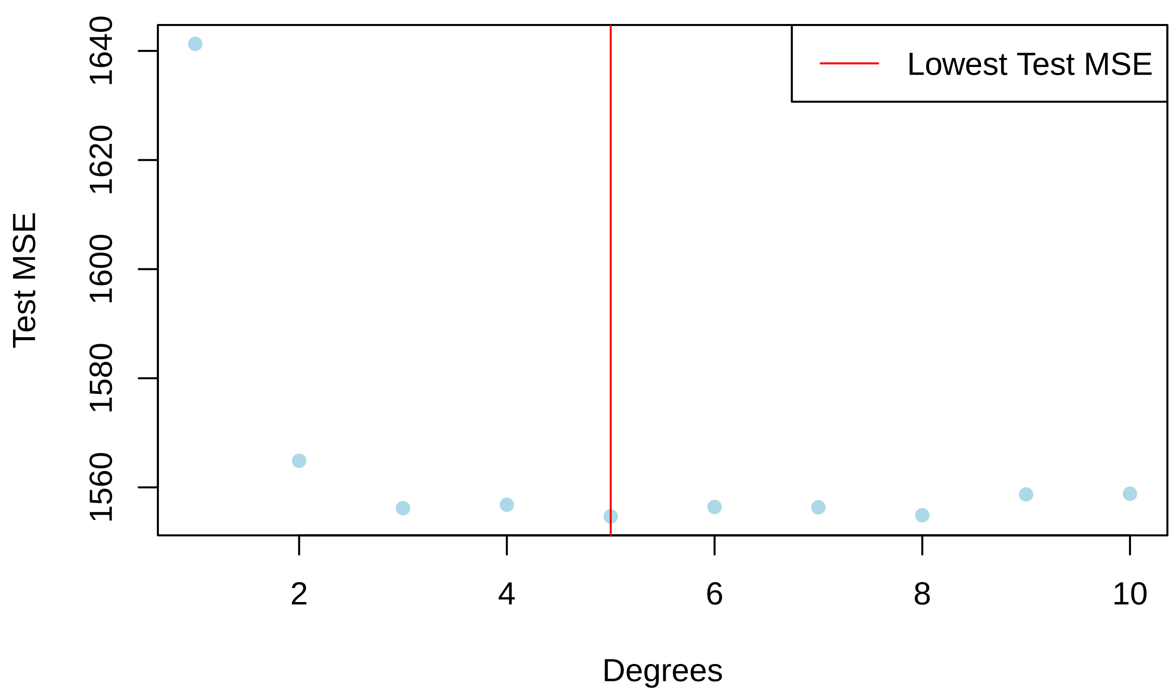
(a) Polynomial

To fit a Polynomial Regression on our train data, we can use 10 fold CV on our train data to see which degree performs the best by comparing their test MSE.

```r
set.seed(435)
Test_MSE <- rep(NA, 10)

for (i in 1:10) {
    fit <- glm(wage ~ poly(age, i), data = wage_train)
    Test_MSE[i] <- cv.glm(wage_train, fit, K = 10)$delta[1]
}


plot(c(1:10), Test_MSE,
     xlab = "Degrees",
     ylab = "Test MSE",
     pch = 16,
     col = "lightblue")
abline(v = which.min(Test_MSE), col = "red")
legend("topright", "Lowest Test MSE", lty = 1, lwd = 1, col = "red")
```



```r
best.fit <- glm(wage ~ poly(age, which.min(Test_MSE)), data = wage_train)
```

- We see that at degree of 5, we have the optimal fit(lowest Test MSE) for Polynomial Regression.
- And now we can go back to the Validation set, we can calculate it's performance(Validation MSE) as:

```r
mean((wage_val$wage - predict(best.fit, data.frame(wage_val)))^2)
```

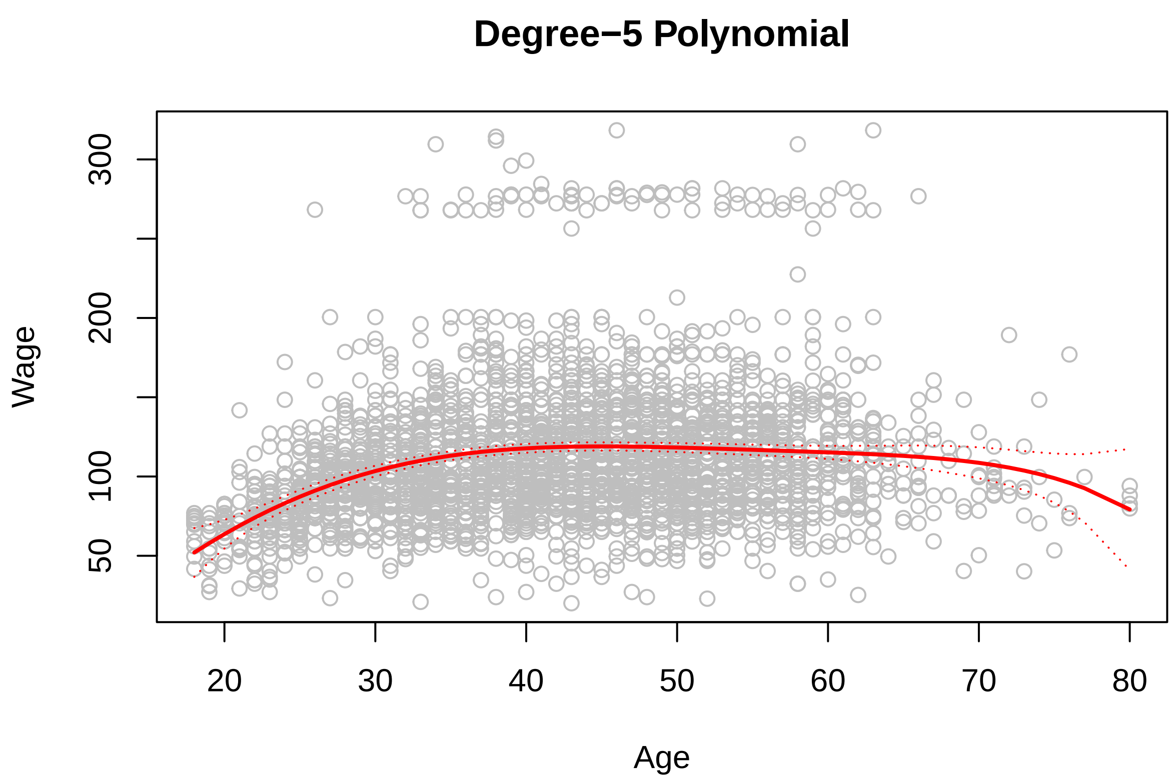
```
## [1] 1714.791
```

- Finally, we can plot our result and see it in a graphical representation alongside with confidence band.

```r
plot(Wage$age, Wage$wage,xlab = "Age", ylab = "Wage", main = "Degree-5 Polynomial", col = "grey")
Wage_new <- Wage %>%
  arrange(age)

preds_poly <- predict(best.fit, data.frame(Wage_new[1:10]), se = TRUE)
se.bands <- cbind(preds_poly$fit + 2 * preds_poly$se.fit,
                  preds_poly$fit - 2 * preds_poly$se.fit)

lines(Wage_new$age, preds_poly$fit,
      lwd = 2, col = "red")
matlines(Wage_new$age, se.bands, lwd = 1, col = "red", lty = 3)
```



- Then, we can use ANOVA to test the null hypothesis that a smaller model M_1 is sufficient to explain the data against the alternative hypothesis that a larger model M_2 is required.

- And we must let M_1, M_2 be nested models: the predictors in M_1 must be a subset of the predictors in M_2.
- Then, we can compare models from lienar to a degree 6 polynomial

```
fit.1 <- lm(wage ~ age, data = wage_train)
fit.2 <- lm(wage ~ poly(age, 2), data = wage_train)
fit.3 <- lm(wage ~ poly(age, 3), data = wage_train)
fit.4 <- lm(wage ~ poly(age, 4), data = wage_train)
fit.5 <- lm(wage ~ poly(age, 5), data = wage_train)
fit.6 <- lm(wage ~ poly(age, 6), data = wage_train)
knitr::kable(cbind("Degree(s)" = c(1:6),anova(fit.1, fit.2, fit.3, fit.4, fit.5, fit.6)))
```

Degree(s)	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	2246	3679457	NA	NA	NA	NA
2	2245	3506486	1	172970.3513	111.4153417	0.0000000
3	2244	3487985	1	18501.8263	11.9175760	0.0005664
4	2243	3483487	1	4497.4494	2.8969407	0.0888865
5	2242	3482845	1	642.4034	0.4137911	0.5201174
6	2241	3479113	1	3731.7207	2.4037121	0.1211889

- First row is showing the results for the smallest model, linear model. It only contains the RSS.
- Second row is comparing linear model with quadratic model and the p-value is essentially zero, indicating a linear fit is not sufficient.
- Third row also indicates that the quadratic model is also not enough, compared to the cubic model.
- Under significance level of 0.05, fourth, fifth, and sixth degree of Polynomial fit does not seem to be necessary.

```
best.fit <- glm(wage ~ poly(age, 3), data = wage_train)
poly_MSE <- mean((wage_val$wage - predict(best.fit, data.frame(wage_val)))^2)
```

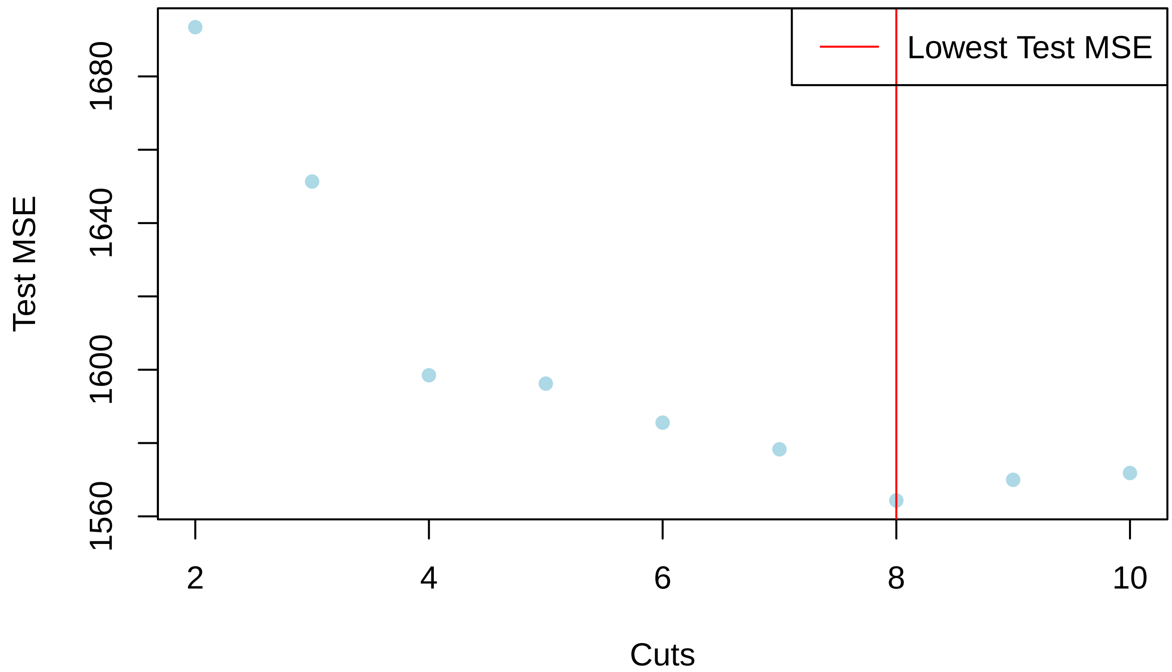
Hence, we would now use third degree of Polynomial fit, and it's Validation MSE is 1717.6166721.

(b) Step Function

- Again, for the step function, we will also perform 10 fold CV on our Training set to see at what level of cuts will give us lowest Test MSE.

```
set.seed(435)
Test_MSE <- rep(NA, 10)
for (i in 2:10) {
    wage_train$age.cut <- cut(wage_train$age, i)
    fit <- glm(wage ~ age.cut, data = wage_train)
    Test_MSE[i] <- cv.glm(wage_train, fit, K = 10)$delta[1]
}

plot(2:10, Test_MSE[-1], xlab = "Cuts", ylab = "Test MSE", pch = 16, col = "lightblue")
abline(v = which.min(Test_MSE), col = "red")
legend("topright", "Lowest Test MSE", lty = 1, lwd = 1, col = "red")
```

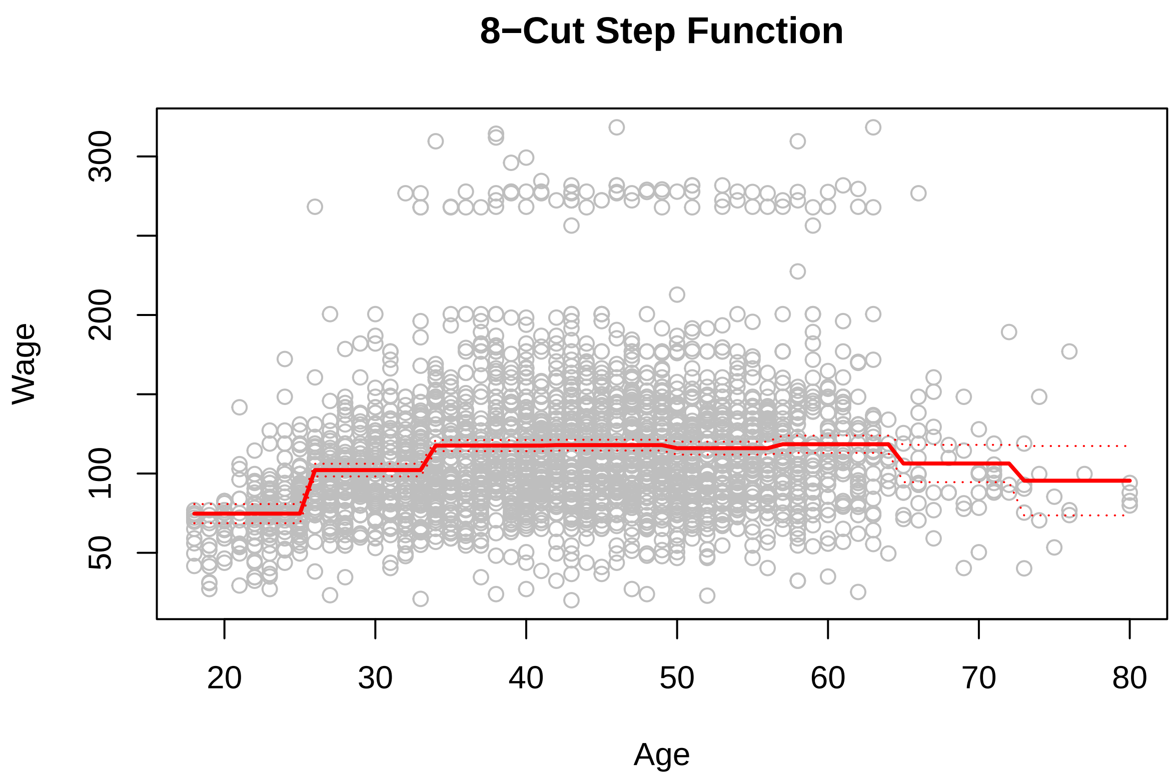


We can see that the Test MSE is at its minimum when we apply 8 cuts for our step function fit. Now, we can plot it and see the fit and it's correspond estimated 95% confidence interval.

```
plot(Wage$age, Wage$wage,xlab = "Age", ylab = "Wage", main = "8-Cut Step Function", col = "grey")
Wage_new <- Wage %>%
  arrange(age) %>%
  mutate(age.cut = cut(age, 8))

wage_train$age.cut <- cut(wage_train$age, 8)
best.fit <- glm(wage ~ age.cut, data = wage_train)
pred_step <- predict(best.fit, data.frame(Wage_new), se = TRUE)
se.bands <- cbind(pred_step$fit + 2 * pred_step$se.fit,
                  pred_step$fit - 2 * pred_step$se.fit)

lines(Wage_new$age, pred_step$fit,
      lwd = 2, col = "red")
matlines(Wage_new$age, se.bands, lwd = 1, col = "red", lty = 3)
```



- And we can calculate it's Validation MSE as:

```
Wage_new <- wage_val %>%
  arrange(age) %>%
  mutate(age.cut = cut(age, 8))
step_MSE <-
  mean((Wage_new$wage - predict(best.fit, data.frame(Wage_new))) ^ 2)
```

Hence, we would now use 8 Cut Step function fit, and it's Validation MSE is 1726.8131328.

(c) Piecewise polynomial

- From part (a), we learned that polynomials that is greater than degree 3 is not need at $\alpha = 0.05$. Hence we can fit a piecewise 3 degree polynomial with knots at ξ. And we can select the number of knots by performing 10 Fold CV on our Training Set.

```
set.seed(435)
Test_MSE <- rep(NA, 10)
wage_train <- wage_train[1:11]
for (i in 2:10) {
    fit <- glm(wage ~ cut(age, i), data = wage_train)
    Test_MSE[i] <- cv.glm(wage_train, fit, K = 10)$delta[1]
}

plot(2:10, Test_MSE[-1], xlab = "Cuts", ylab = "Test MSE", pch = 16, col = "lightblue")
abline(v = which.min(Test_MSE), col = "red")
legend("topright", "Lowest Test MSE", lty = 1, lwd = 1, col = "red")
```

d.

```r
fit <- lm(wage ~ bs(age, knots = c(25, 40, 60)), data = wage_train)
```

e.

```r
fit.smooth_spline <- smooth.spline(wage_train$age, wage_train$wage, cv = TRUE)
```

5.

```r
# Remove the name column
set.seed(435)
Auto_new <- Auto %>%
  select(-name)

train <- sample(1 : nrow(Auto_new), nrow(Auto_new) / 2)
```

a.

- Our unpruned tree looks like:

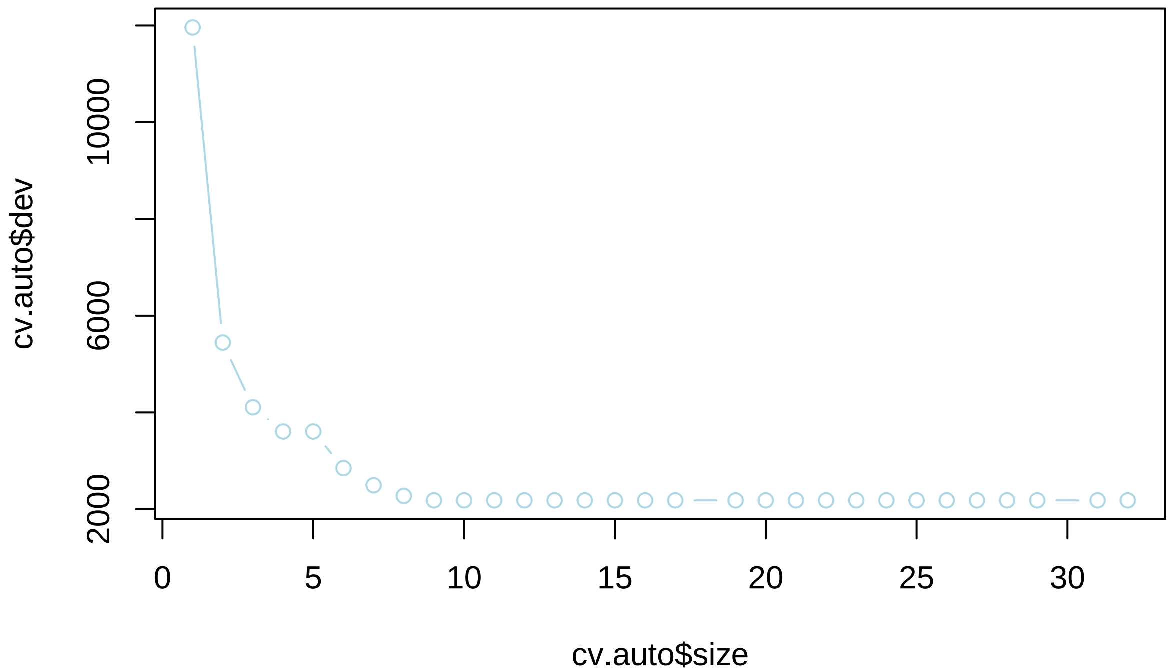
```r
tree.auto <- tree(mpg~ .,Auto_new, subset = train,
                  control = tree.control(nobs = length(train),
                                         mindev = 0))
```

- And to prune the tree, we can perform 10 Fold Cross Validation

```r
set.seed(435)

cv.auto <- cv.tree(tree.auto, K = 10)

plot(cv.auto$size, cv.auto$dev, type = "b", col =  "lightblue")
```



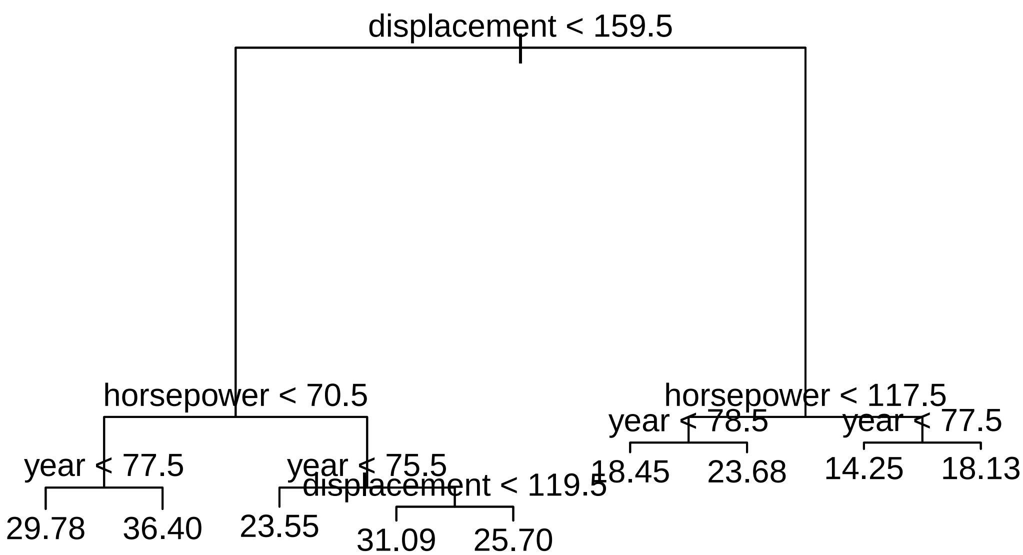

```
knitr::kable(tibble("number of terminal nodes" = cv.auto$size,
             "Test RSS" = cv.auto$dev))
```

number of terminal nodes	Test RSS
32	2182.568
31	2182.568
29	2182.568
28	2182.568
27	2182.568
26	2182.568
25	2182.568
24	2182.568
23	2182.568
22	2182.568
21	2182.568
20	2182.568
19	2182.568
17	2182.568
16	2182.568
15	2182.568
14	2182.568
13	2182.568
12	2182.568
11	2182.568
10	2182.568
9	2182.568
8	2274.428
7	2494.198
6	2849.457
5	3605.969

number of terminal nodes	Test RSS
4	3605.969
3	4106.249
2	5444.325
1	11959.436

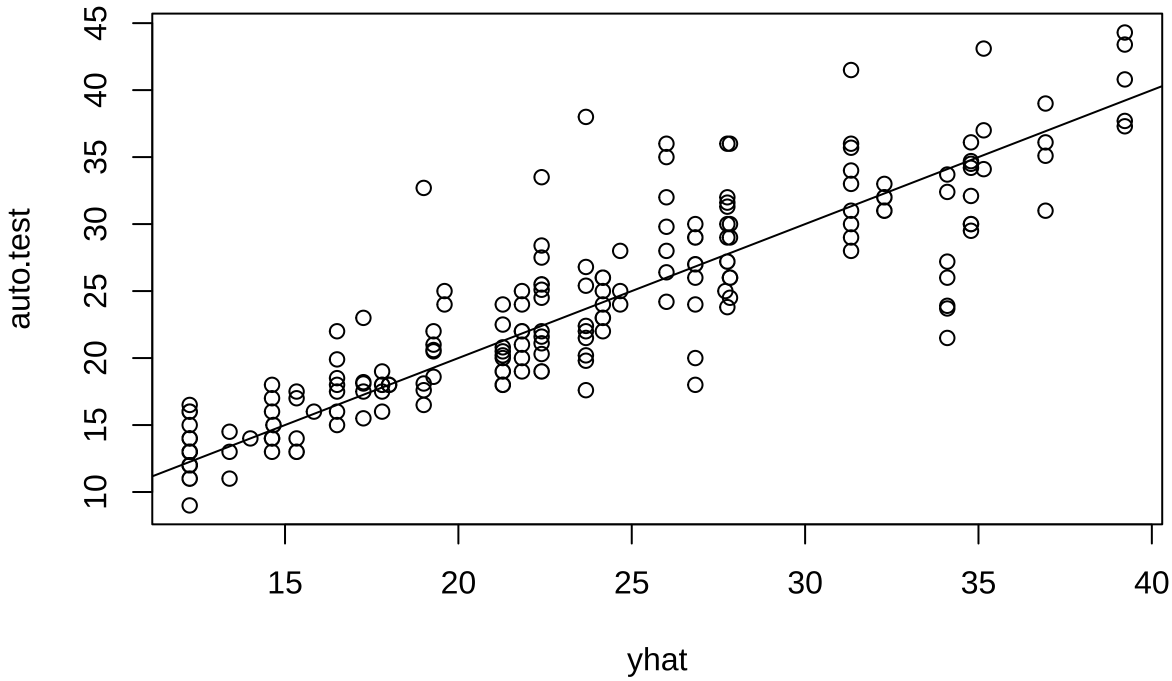
Hence, we shall pick the number of terminal nodes with lowest test RSS, and we can see that after nodes with 9, the Test RSS does not decrease anymore, hence we shall prune our tree to 9 terminal nodes.

```
prune.auto <- prune.tree(tree.auto, best = 9)
plot(prune.auto)
text(prune.auto, pretty = 0)
```




```
yhat <- predict(tree.auto, newdata = Auto_new[-train, ])
auto.test <- Auto_new[-train, "mpg"]

plot(yhat, auto.test)
abline(0, 1)
```




```
Test_MSE <- mean((yhat - auto.test)^2)
```

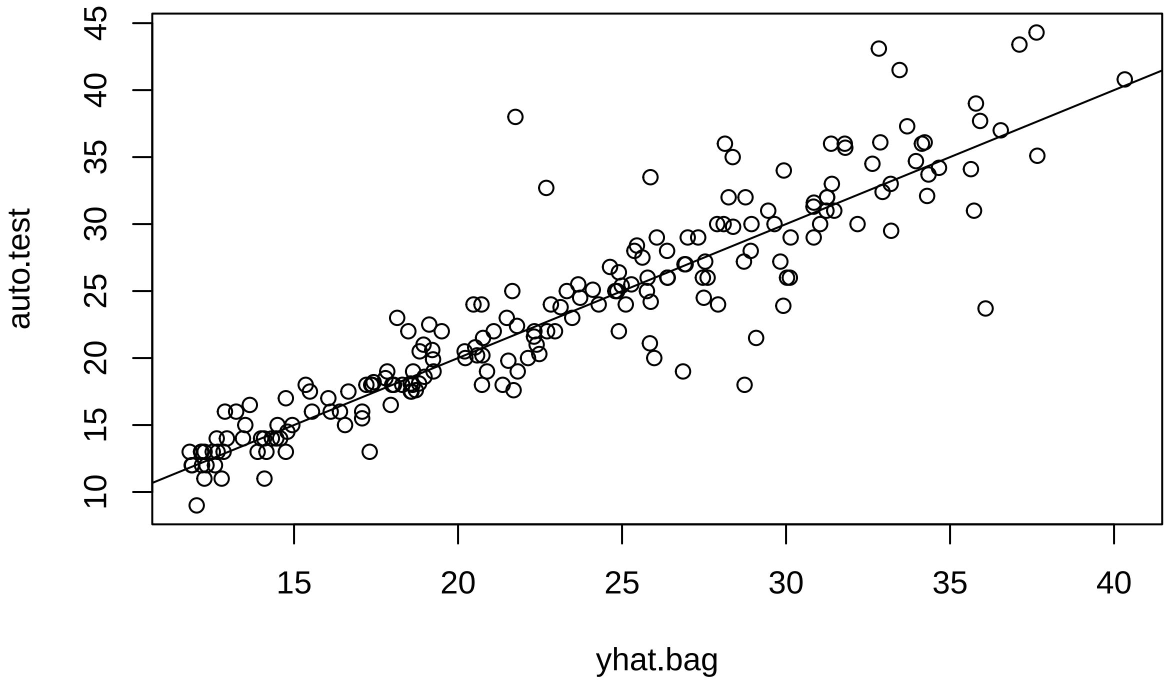
And my test MSE is 13.6200007 for my regression tree to predict a car's gas mileage.

b

- Since Bagging is just a special case of RandomForest, we can perform it by setting $mtry = p$.

```
set.seed(435)
bag.auto <- randomForest(mpg ~ ., data = Auto_new, subset = train, mtry = 7, importance = TRUE)

yhat.bag <- predict(bag.auto, newdata = Auto_new[-train, ])
plot(yhat.bag, auto.test)
abline(0, 1)
```




```r
mean((yhat.bag - auto.test)^2)
```

```
## [1] 9.757461
```

And our Test MSE is 9.7574607.