homework4

Sunny Lee

2/23/2021

1)

```
library("MASS")
data("Boston")
```

```
data <- Boston[, c("medv","crim")]
train <- data[-1, ]
test <- data[1, ]
```

2)

```
x <- data$medv
y <- data$crim
x_i <- x[-1]
y_i <- y[-1]
model <- lm(y_i~x_i)
y_pred <- predict(model, data.frame(x_i = x[1]))
```

3)

```
error <- (y[1]-y_pred)^2
```

4)

```
library(boot)
mse <- c(1:nrow(data))
for (i in 1:nrow(data)){
  x <- data$medv
  y <- data$crim
  x_i <- x[-i]
  y_i <- y[-i]
  model <- glm(y_i~x_i)
  mse[i] <- (y[i] - predict(model, data.frame(x_i = x[i])))^2
}
```

5)

```
mean(mse)
```

```
## [1] 63.5012
```

```
#using boot library to check our CV
a <- glm(crim~medv, data = data)
cva <- cv.glm(data, a)$delta[1]
```

6)

```
mse1 <- c(1:nrow(data))
for (i in 1:nrow(data)){
  x <- data$medv
  y <- data$crim
  x_i <- x[-i]
  y_i <- y[-i]
  model <- glm(y_i~poly(x_i, 2))
  mse1[i] <- (y[i] - predict(model, data.frame(x_i = x[i])))^2
}
```

For this model, we find that the $CV_{(n)}$ is 48.45057, which is much lower than it is for our previous linear regression model.

```
mean(mse1)
```

```
## [1] 48.45057
```

```
#using boot library to check our CV
b <- glm(crim~poly(medv, 2), data = data)
cvb <- cv.glm(data, b)$delta[1]
```

7)

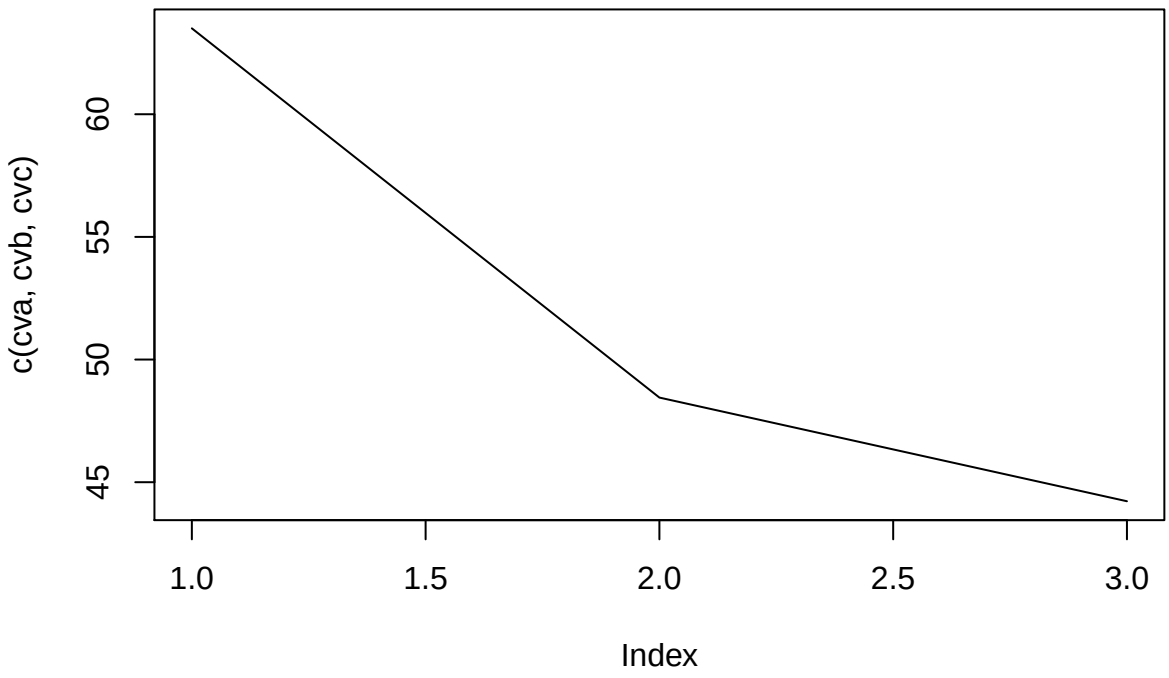
```
mse2 <- c(1:nrow(data))
for (i in 1:nrow(data)){
  x <- data$medv
  y <- data$crim
  x_i <- x[-i]
  y_i <- y[-i]
  model <- glm(y_i~poly(x_i, 3))
  mse2[i] <- (y[i] - predict(model, data.frame(x_i = x[i])))^2
}
mean(mse2)
```

```
## [1] 44.22393
```

```
#using boot library to check our CV
c <- glm(crim~poly(medv, 3), data = data)
cvc <- cv.glm(data, c)$delta[1]
```

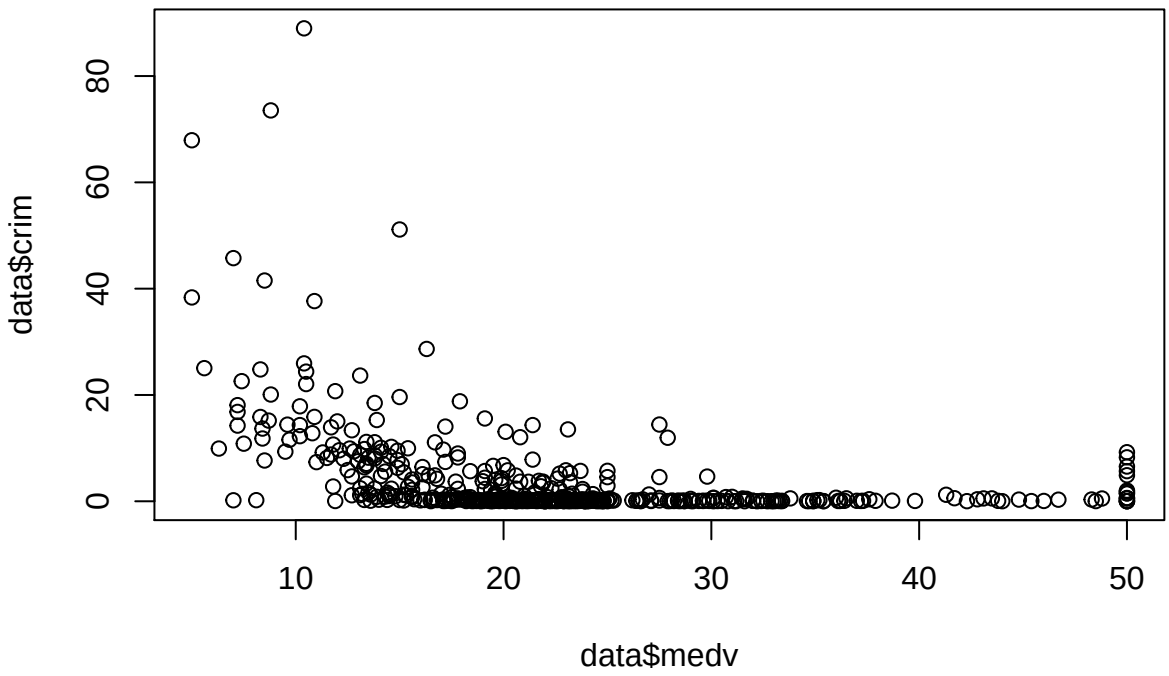
8)

```
plot(c(cva, cvb, cvc), type = "l")
```



9) Looking at the graph above, we find that the polynomial regression of degree 3 had the lowest LOOCV value. By looking at the graph below:

```
plot(data$crim~data$medv)
```



we see the relationship between medv and crim is closer to an exponential decay, meaning we would generally see a better MSE score the higher the degree we choose.

10)

```
summary(a)
```

```
##
## Call:
## glm(formula = crim ~ medv, data = data)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -9.071  -4.022  -2.343   1.298  80.957
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.79654    0.93419   12.63   <2e-16 ***
## medv        -0.36316    0.03839   -9.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 62.95551)
##
##     Null deviance: 37363  on 505  degrees of freedom
## Residual deviance: 31730  on 504  degrees of freedom
## AIC: 3536
```

```
##
## Number of Fisher Scoring iterations: 2
```

```
summary(b)
```

```
##
## Call:
## glm(formula = crim ~ poly(medv, 2), data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -18.802   -3.127   -0.593    2.031   75.204
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.6135     0.3069   11.78   <2e-16 ***
## poly(medv, 2)1 -75.0576     6.9033  -10.87   <2e-16 ***
## poly(medv, 2)2  88.0862     6.9033   12.76   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 47.65487)
##
##     Null deviance: 37363  on 505  degrees of freedom
## Residual deviance: 23970  on 503  degrees of freedom
## AIC: 3396.1
##
## Number of Fisher Scoring iterations: 2
```

```
summary(c)
```

```
##
## Call:
## glm(formula = crim ~ poly(medv, 3), data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -24.427   -1.976   -0.437    0.439   73.655
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.614      0.292  12.374  < 2e-16 ***
## poly(medv, 3)1  -75.058      6.569 -11.426  < 2e-16 ***
## poly(medv, 3)2   88.086      6.569  13.409  < 2e-16 ***
## poly(medv, 3)3  -48.033      6.569  -7.312 1.05e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 43.15376)
##
##     Null deviance: 37363  on 505  degrees of freedom
## Residual deviance: 21663  on 502  degrees of freedom
## AIC: 3346.9
##
## Number of Fisher Scoring iterations: 2
```

Looking at the significance of the coefficients, we find the coefficients are all very important to fitting the data for each degree polynomial. This results match what we see with the CV score, as we are adding more variables which are significant in predicting crim.