Datamining_HW1

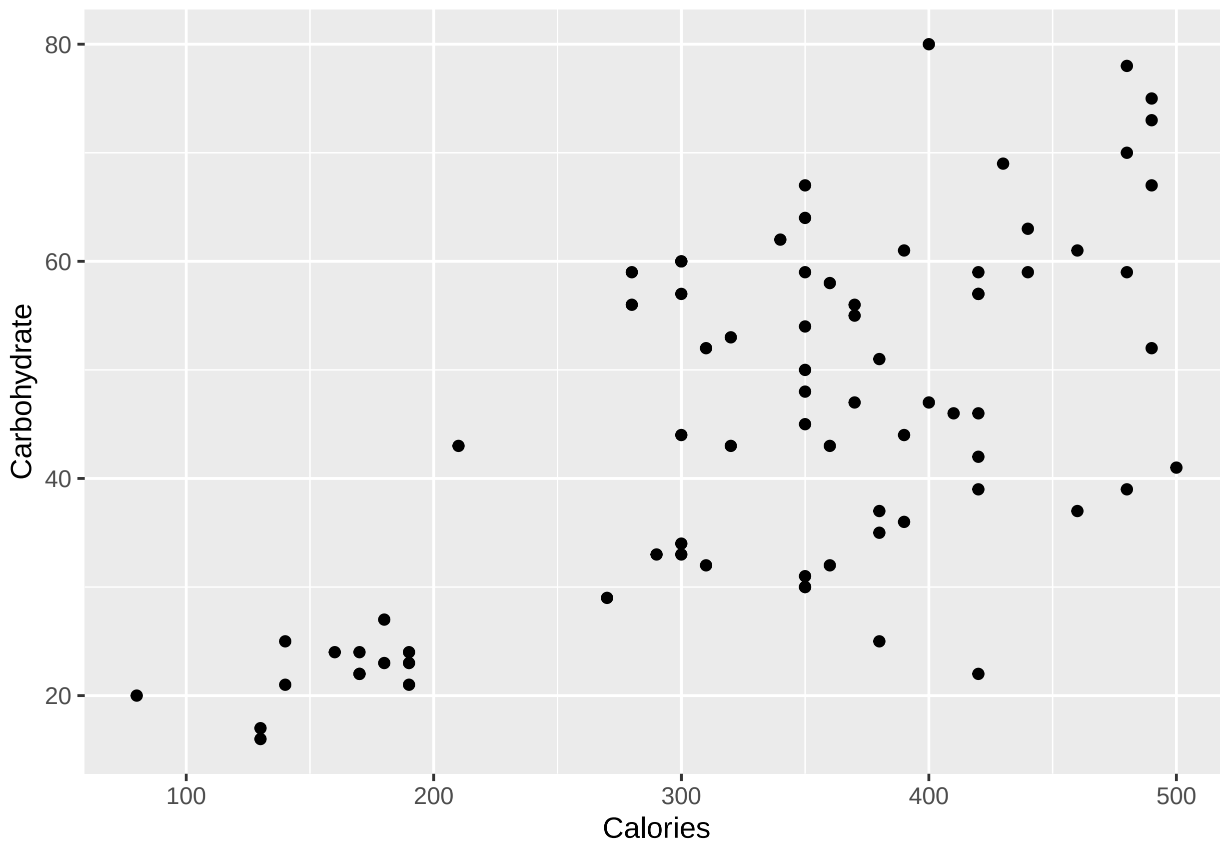
김민국(2014-12512)

2019/10/6

Problem 1

1-1

```
data("starbucks")
attach(starbucks)
ggplot(data = starbucks,
       aes(x=calories, y=carb)) + geom_point() +
  labs(x = "Calories", y="Carbohydrate") + labs(x = "Calories", y = "Carbohydrate")
```



-> Carlories와 Carbohydrate 사이에는 양의 상관관계가 있다고 볼 수 있다. Carlories의 값이 300이상인 곳들에 데이터들이 주로 모여있다.

1-2

```r
starbucks_lm <- lm(carb ~ calories)
starbucks_lm
```

```
##
## Call:
## lm(formula = carb ~ calories)
##
## Coefficients:
## (Intercept)     calories
##       8.944        0.106
```

1-3

```r
starbuks_lm_coef <- starbucks_lm$coefficients
expression <- paste("Carbohydrate =",
                     paste(starbuks_lm_coef[1],
                           paste("Carlories",starbuks_lm_coef[2],sep = " * "),
                           "error"
                           ,sep =  " + "))
expression
```

```
## [1] "Carbohydrate = 8.94356047663464 + Carlories * 0.10603088705631 + error"
```

-> Carolies가 0일 때 Carbohydrare는 8.9436정도인 intercept 값을 갖게되며 Carlories가 1씩 증가할 때마다 Carbohydrate는 약 0.1060 씩 증가하게 된다.

1-4

```r
summary(starbucks_lm)
```

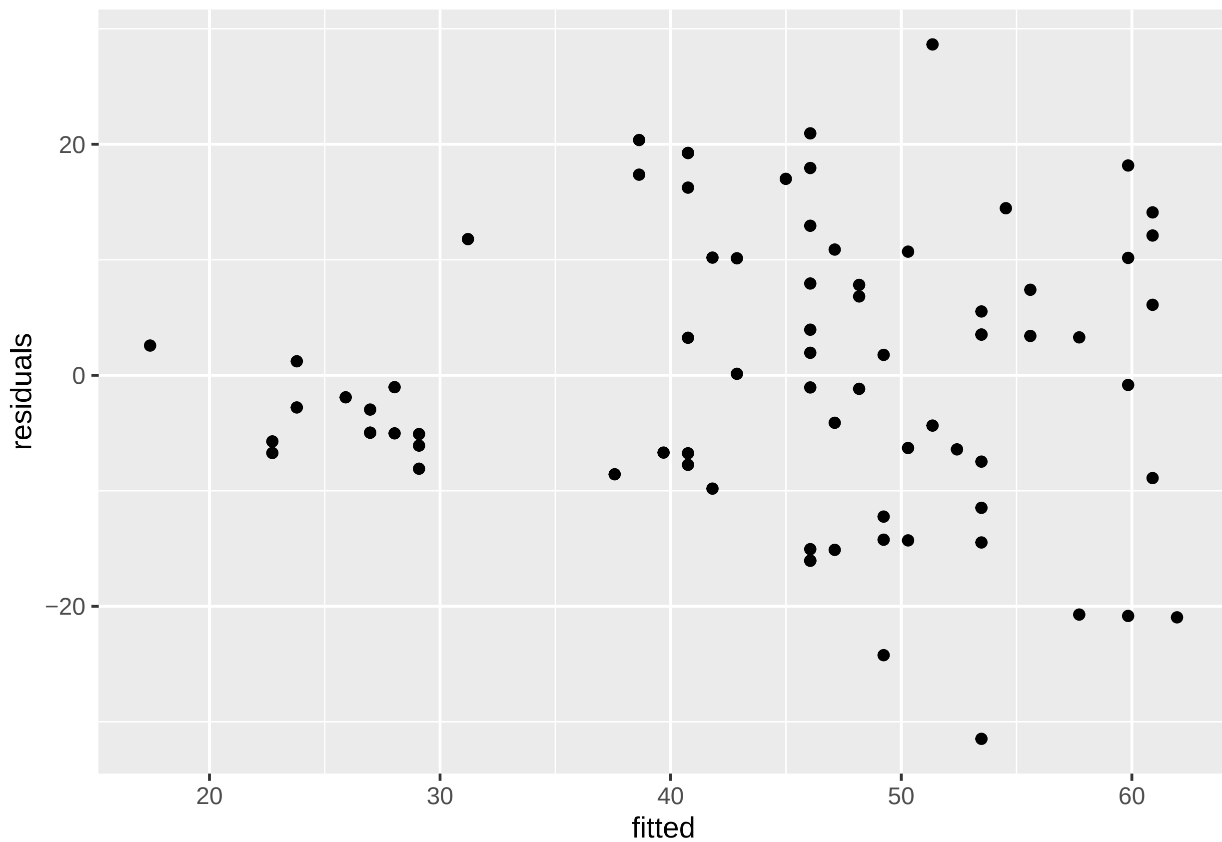
```
##
## Call:
## lm(formula = carb ~ calories)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -31.477  -7.476  -1.029  10.127  28.644
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.94356    4.74600   1.884   0.0634 .
## calories     0.10603    0.01338   7.923 1.67e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 12.29 on 75 degrees of freedom
## Multiple R-squared:  0.4556, Adjusted R-squared:  0.4484
## F-statistic: 62.77 on 1 and 75 DF,  p-value: 1.673e-11
```

-> R^2값은 0.4556, adjusted R^2는 0.4484 이므로 해당 모형이 45%정도 설명할 수 있다고 생각할 수 있다.

1-5

```r
starbuks_lm_1 <- fortify(starbucks_lm)
ggplot(starbuks_lm_1, aes(x = .fitted, y=.resid)) +
  geom_point() + labs(x="fitted",y="residuals")
```




```r
detach(starbucks)
```

-> 0을 기준으로 fitted값이 커질수록 분산이 커지는 것이 확인된다, 등분산성을 만족하지 않는다고 생각할 수 있다.
-> 0을 기준으로 잔차들이 랜덤하게 분포한다고 볼 수도 있다. 선형성을 어느정도는 만족할 수 있다고 생각할 수 있다.

Problem 2

2-1

```r
absent <- read.csv("absenteeism.csv")
absent_1 <- absent
absent_1$eth <- as.numeric(absent_1$eth) - 1
absent_1$sex <- as.numeric(absent_1$sex) - 1
absent_1$lrn <- as.numeric(absent_1$lrn) - 1
head(absent_1)
```

```
##   X eth sex age lrn days
## 1 1   0   1  F0   1    2
## 2 2   0   1  F0   1   11
## 3 3   0   1  F0   1   14
## 4 4   0   1  F0   0    5
## 5 5   0   1  F0   0    5
## 6 6   0   1  F0   0   13
```

2-2

```r
absent_lm <- lm(days ~ eth + sex + lrn, data = absent_1)
absent_lm
```

```
##
## Call:
## lm(formula = days ~ eth + sex + lrn, data = absent_1)
##
## Coefficients:
## (Intercept)          eth          sex          lrn
##      18.932       -9.112        3.104        2.154
```

2-3

```r
absent_lm_coef <- absent_lm$coefficients
expression1 <- paste("Days =", paste(absent_lm_coef[1],
                                     paste("Eth",round(absent_lm_coef[2],4),sep = " * "),
                                     paste("Sex",round(absent_lm_coef[3],4),sep = " * "),
                                     paste("lrn",round(absent_lm_coef[4],4),sep = " * "),
                                     "e",sep =  " + "))
expression1
```

```
## [1] "Days = 18.93184820771 + Eth * -9.1122 + Sex * 3.1043 + lrn * 2.1542 + e"
```

-> 설명변수들이 모두 0의 값을 가질때는 Days는 intercept항인 18.9318정도의 값을 가지게 된다.
-> 설명 변수 하나의 값이 변하게 될 때 나머지 설명 변수 2개는 값이 고정된다고 가정하자.
-> Ethnicity가 aboriginal(0)에서 Not aboriginal(1)로 변하게 되면 Days는 9.1122정도 감소하게 된다.

-> Sex가 female(0)에서 male(1)로 변하게 되면 Days는 3.1043정도 증가하게 된다.

-> Learning ablility가 average(0)에서 slow learner(1)로 변하게 되면 Days는 2.1542정도 증가하게 된다.

2-4

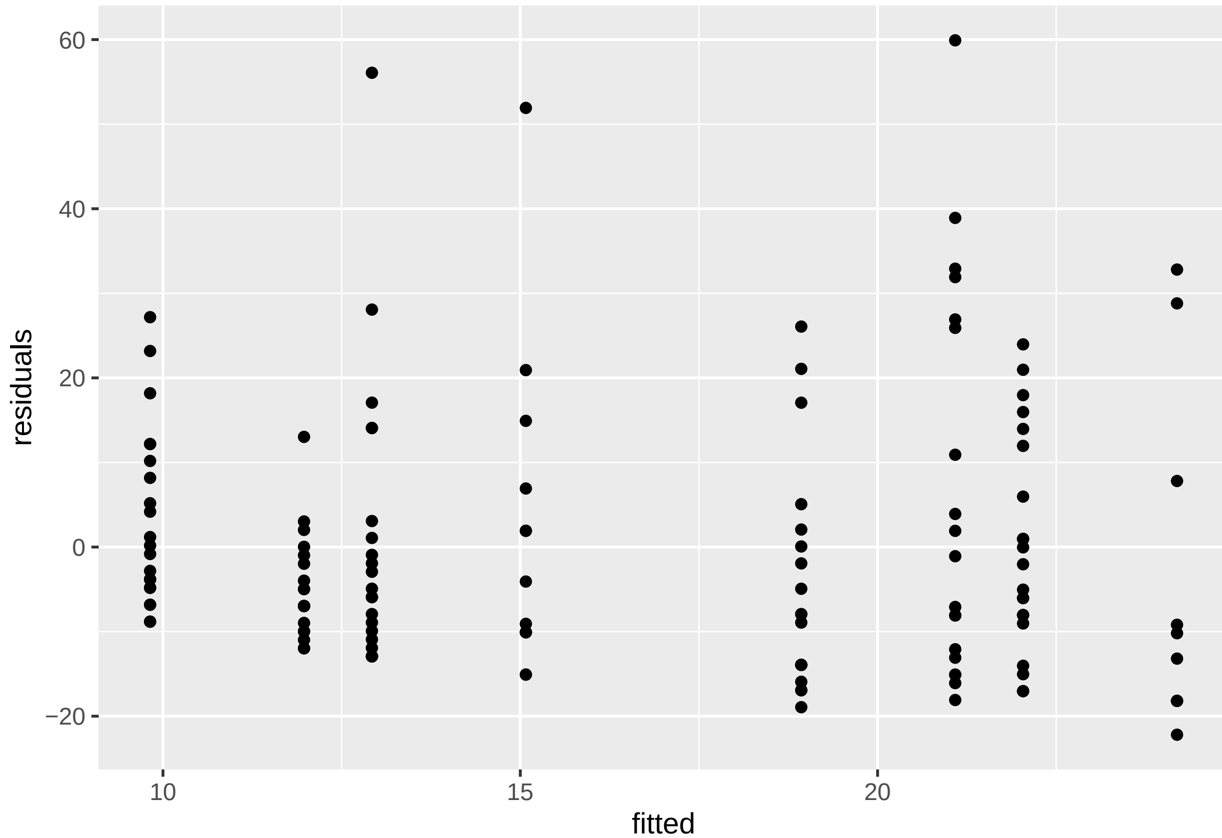
```
summary(absent_lm)
```

```
##
## Call:
## lm(formula = days ~ eth + sex + lrn, data = absent_1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.190 -10.078  -4.928   5.768  59.914
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   18.932      2.570   7.365 1.32e-11 ***
## eth           -9.112      2.599  -3.506 0.000609 ***
## sex            3.104      2.637   1.177 0.241108
## lrn            2.154      2.651   0.813 0.417732
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.67 on 142 degrees of freedom
## Multiple R-squared:  0.08933,    Adjusted R-squared:  0.07009
## F-statistic: 4.643 on 3 and 142 DF,  p-value: 0.003967
```

-> adjusted R^2 값은 0.07009로 매우 작다. 즉 모델이 7%정도밖에 설명력을 갖지 못한다.

2-5

```
absent_lm_1 <- fortify(absent_lm)
ggplot(absent_lm_1, aes(x = .fitted, y=.resid)) +
  geom_point() + labs(x="fitted",y="residuals")
```



-> 잔차들이 random하다고 보기 힘들다. 선형성과 등분산성을 모두 확인하기 힘들다고 볼 수 있다.

2-6

```
newdata <- data_frame(eth = c(1,1,1,0,0), sex = c(0,1,0,1,0), lrn = c(0,0,1,1,0))
```

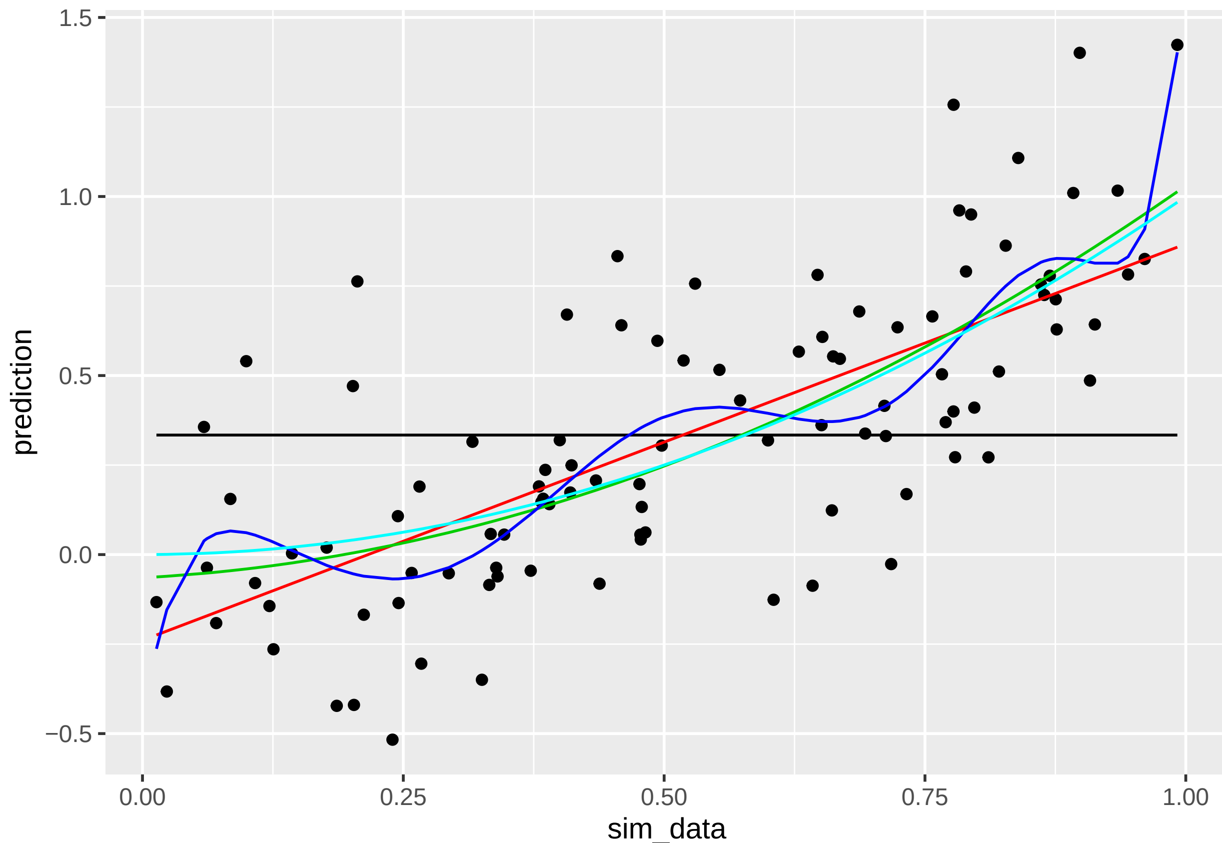
```
## Warning: `data_frame()` is deprecated, use `tibble()`.
## This warning is displayed once per session.
```

```
predict_day <- predict(absent_lm, newdata = newdata)
newdata_1 <- cbind(newdata,predict_day)
newdata_1
```

```
##   eth sex lrn predict_day
## 1   1   0   0    9.819607
## 2   1   1   0   12.923862
## 3   1   0   1   11.973764
## 4   0   1   1   24.190261
## 5   0   0   0   18.931848
```

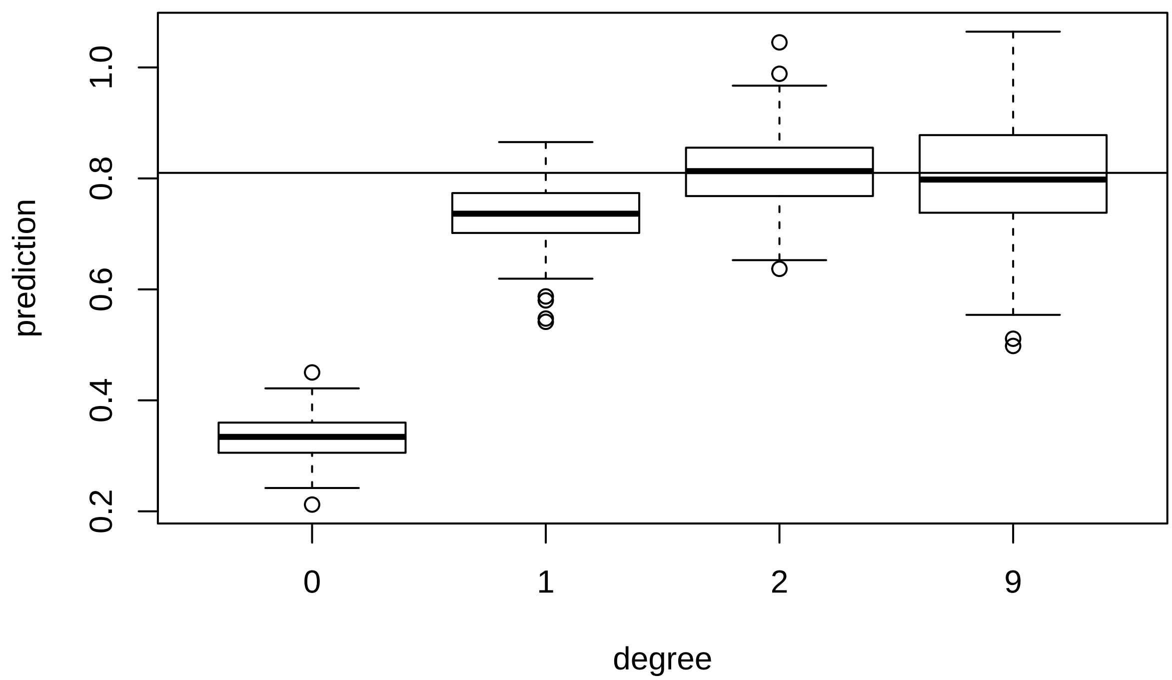
Problem3

```r
f <- function(x) { x^2 }
get_sim_data <- function(f, sample_size = 100) {
  x = runif(n = sample_size, min = 0, max = 1)
  y = rnorm(n = sample_size, mean = f(x), sd = 0.3)
  data.frame(x, y)
}

set.seed(1)
sim_data <- get_sim_data(f) #simulation data 생성
lm1 <- lm(y ~ 1, data = sim_data)
lm2 <- lm(y ~ x, data = sim_data)
lm3 <- lm(y ~ poly(x, degree = 2), data = sim_data)
lm4 <- lm(y ~ poly(x, degree = 9), data = sim_data)

sim_data_1 <- data.frame(x = sim_data$x, y = sim_data$y,
                         lm1 = lm1$fitted.values,
                         lm2 = lm2$fitted.values,
                         lm3 = lm3$fitted.values,
                         lm4 = lm4$fitted.values,
                         x2 = (sim_data$x)^2)
ggplot(data = sim_data_1) + geom_point(aes(x=x, y=y)) +
  geom_line(aes(x = x, y = lm1),color = 1) +
  geom_line(aes(x = x, y = lm2),color=2) +
  geom_line(aes(x = x, y = lm3),color=3) +
  geom_line(aes(x = x, y = lm4),color=4) +
  geom_line(aes(x = x, y = x2),color=5, show.legend = T) +
  labs(x = "sim_data", y = "prediction")
```




```
n_sims <- 250
n_models <- 4
df <- data.frame(0.90)
colnames(df) <- 'x'
r <- data.frame(NA)
for(sim in 1:n_sims){
  set.seed(sim)
  sim_data <- get_sim_data(f)
  lm1 <- lm(y ~ 1, data = sim_data)
  lm2 <- lm(y ~ x, data = sim_data)
  lm3 <- lm(y ~ poly(x, degree = 2), data = sim_data)
  lm4 <- lm(y ~ poly(x, degree = 9), data = sim_data)
  r[sim,1] <- predict(lm1, newdata = df)
  r[sim,2] <- predict(lm2, newdata = df)
  r[sim,3] <- predict(lm3, newdata = df)
  r[sim,4] <- predict(lm4, newdata = df)
}
colnames(r)<- c('0','1','2','9')
boxplot(r, xlab = "degree", ylab = "prediction")
```

```r
abline(h = 0.9 ^ 2)
```



-> Degree가 증가할수록 박스의 크기는 커지고, 최대값 최소값의 간격이 멀다. 즉 variance가 커지고 있음을 확인할 수 있다.

-> Degree가 증가할수록 중앙값이 True value에 가까운 것을 확인할 수 있다. 즉, bias가 줄어들고 있음을 확인할 수 있다.

-> 즉, bias를 낮추기 위해 degree를 증가시키면 그만큼 variance가 증가하는 Bias-Variance tradeoff를 확인할 수 있다.

-> 이 모형에서는 degree = 2일 때가 degree = 9 일 때보다 bias와 variance가 작은 것을 확인할 수 있다.