hw1_4630

Akhil Havaldar (ash2sfp)

9/4/2022

Q5

```
#a)
gety <- function(x,intercept,slope,eps.sigma)

{
  y <- intercept + slope*(x^2) + rnorm(length(x),0,eps.sigma)
  return(y)
}


##generate the values of x
x<-rep(seq(1,10,1),20)

##initialize values for simulation
beta0 <- 0 ##intercept
beta1 <- 2 ##slope
sig <- 2 ##sd of error term

#b)
reps <- 100000
store.y<-array(0,reps)
set.seed(4630)
y0 <- array(0,reps)
yhat <- array(0,reps)

for (i in 1:reps)

{
  y<-gety(x, intercept=beta0, slope=beta1, eps.sigma=sig)
  store.y[i] <- y
  model <- lm(y~x)
  y2<-gety(7, intercept=beta0, slope=beta1, eps.sigma=sig)
  y0[i] <- y2
  yhat[i] <- model$coeff[2]*7 + model$coeff[1]


}
```

```r
#c)
mse_lhs <- mean((y0-yhat)^2)
print(mse_lhs)
```

```
## [1] 148.1069
```

```r
#d)
avgpred <- mean(yhat)
print(avgpred)
```

```
## [1] 109.9993
```

```r
#e)
var <- mean((yhat-avgpred)^2)
print(var)
```

```
## [1] 0.02531016
```

```r
new <- 2 * (7^2)

bias <- (new - avgpred)^2
print(bias)
```

```
## [1] 143.9826
```

```r
errorvar <- mean((y0 - new)^2)
print(errorvar)
```

```
## [1] 4.015478
```

```r
rhs <- var + bias + errorvar
print(rhs)
```

```
## [1] 148.0234
```

```r
#g)

print(mse_lhs-rhs)
```

f) it should be close to 4 since the SD of the error is 2, so the variance would be 2^2 which equals 4.

```
## [1] 0.08356216
```

difference is extremely small between the LHS and RHS of the equation

FIRST PART OF HW

Q1

```
x <- c(70, 75, 80, 80, 85, 90)
y <- c(75, 82, 80, 86, 90, 91)

model <- lm(y~x)
#a)
summ <- summary(model)
slope1 <- summ$coeff[2] #slope
print(slope1)
```

```
## [1] 0.8
```

```
int1 <- summ$coeff[1]   # intercept
print(int1)
```

```
## [1] 20
```

```
#b)
mse <- mean(summ$residuals^2)
print(mse)
```

```
## [1] 5
```

```
#c)
xtrain <- x[-6]
ytrain <- y[-6]
model2 <- lm(ytrain~xtrain)
summ2 <- summary(model2)

slope2 <- summ2$coeff[2]    # slope
print(slope2)
```

```
## [1] 0.8923077
```

```
int2 <- summ2$coeff[1]      # intercept
print(int2)
```

```
## [1] 13
```

```
msetrain <- mean(summ2$residuals^2)
print(msetrain)
```

```
## [1] 5.538462
```

```r
xtest <- x[6]
ytest <- y[6]
f <- slope2*xtest + int2
g <- sapply(ytest, "-", f)
msetest <- mean(g ^2)
print(msetest)
```

```
## [1] 5.325444
```

they would not stay the same

Q2

```r
x
```

```
## [1] 70 75 80 80 85 90
```

```r
y
```

```
## [1] 75 82 80 86 90 91
```

```r
xnew <- 78
xy <- as.data.frame(cbind(y,x))

xy['mink'] <- (abs(x-xnew)^2)^(1/2)

# for K = 3
mink3 <- head(xy[order(xy$mink),], n = 3)
minkpred3 <- mean(mink3$y)
print(minkpred3)
```

```
## [1] 82.66667
```

```r
# for K = 4
mink4 <- head(xy[order(xy$mink),], n = 4)
minkpred4 <- mean(mink4$y)
print(minkpred4)
```

```
## [1] 84.5
```

Q3

a) flexible is worse (flexible is better with large sample sizes; A flexible model will cause overfitting because of the small sample size. This usually means a bigger inflation in variance and a small reduction in bias.)

b) flexible is better (flexible is better to find nonlinear effect)

c) flexible is worse (flexible model will capture too much of the noise in the data due to the large variance of the errors.)

Q4

#a) inflexible #b) inflexible #c) flexible