Yichen Dong Module 8 HW

Yichen Dong

October 25, 2018

## Problem 1

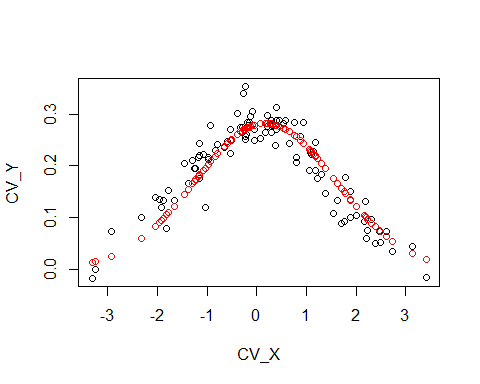
### Part A

CV\_X = scan("CV\_X.txt")  
CV\_Y = scan("CV\_Y.txt")  
mean\_CV = mean(CV\_X)  
mean\_CV

## [1] 0.1734736

Our MLE for mu is the mean of the dataset, or .1734. This means that we have a normal pdf 1/sqrt(4\*pi)\*exp(-(x-.1734)2/4)

CV = as.data.frame(cbind(CV\_X,CV\_Y))  
CV = CV %>%  
 mutate(normal = dnorm(CV\_X,mean\_CV,sqrt(2)))%>%  
 mutate(error = abs(CV\_Y-normal))  
plot(CV\_X,CV\_Y)  
points(CV\_X,CV$normal,col = "red")

 Here we have the points of the data along with the predicted values based on a N(.1734,2) distribution.

sum(CV$error)/length(CV$CV\_X)

## [1] 0.02518254

This is our apparent error.

### Part B and C

CV = CV %>%  
 mutate(group = floor((as.numeric(rownames(CV))-1)/50))  
mean\_CV0 = mean(CV$CV\_X[CV$group == 0])  
mean\_CV1 = mean(CV$CV\_X[CV$group == 1])  
CV = CV %>%  
 mutate(norm\_group\_0 = dnorm(CV\_X,mean\_CV1,sqrt(2))  
 ,norm\_group\_1 = dnorm(CV\_X,mean\_CV0,sqrt(2)))  
CV = CV%>%  
 mutate(error\_0 = ifelse(group == 0, abs(CV\_Y - norm\_group\_0),0)  
 ,error\_1 = ifelse(group == 1, abs(CV\_Y - norm\_group\_1),0))   
(sum(CV$error\_0)+sum(CV$error\_1))/length(CV$CV\_X)

## [1] 0.03103595

This is our cross validated error using two halves of the dataset. As we can see, our second value was larger than the first. This means that our apparent error likely underpredicted the actual error that would appear. By using balnced half sampling, we are able to get closer to the true error that would occur with fitting the data.

## Problem 2

## Problem 3

### Part b

jackknife = scan("Jackknife.txt")  
jackknife = as.data.frame(jackknife)  
jack\_mean = mean(jackknife$jackknife)  
b2 = sum((jackknife$jackknife -jack\_mean)^4)/sum((jackknife$jackknife -jack\_mean)^2)^2  
  
k=5  
groups = length(jackknife$jackknife)/k  
jack\_group = rep(1:groups,each =k)  
jackknife= cbind.data.frame(jackknife,jack\_group)  
T\_minus\_j = NULL  
  
for(i in 1:groups){  
 jk\_minus\_j = jackknife$jackknife[jackknife$jack\_group != i]  
 jk\_minus\_j\_mean = mean(jk\_minus\_j)  
 T\_minus\_j[i] = sum((jk\_minus\_j -jk\_minus\_j\_mean)^4)/sum((jk\_minus\_j -jk\_minus\_j\_mean)^2)^2  
}  
  
T\_bar\_dot = mean(T\_minus\_j)  
J\_T = groups\*b2 - (groups-1)\*T\_bar\_dot  
  
jk\_var = (groups-1)/groups \* sum((T\_minus\_j - T\_bar\_dot)^2)  
paste("Values for k=",k,"; b2:",round(b2,4),", J\_T:",round(J\_T,6),", T\_bar\_dot:",round(T\_bar\_dot,4),"SD", round(sqrt(jk\_var),4))

## [1] "Values for k= 5 ; b2: 0.0267 , J\_T: 1e-04 , T\_bar\_dot: 0.0281 SD 0.0037"

k=1  
groups = length(jackknife$jackknife)/k  
jack\_group = rep(1:groups,each =k)  
jackknife= cbind.data.frame(jackknife,jack\_group)  
T\_minus\_j = NULL  
  
for(i in 1:groups){  
 jk\_minus\_j = jackknife$jackknife[jackknife$jack\_group != i]  
 jk\_minus\_j\_mean = mean(jk\_minus\_j)  
 T\_minus\_j[i] = sum((jk\_minus\_j -jk\_minus\_j\_mean)^4)/sum((jk\_minus\_j -jk\_minus\_j\_mean)^2)^2  
}  
  
T\_bar\_dot = mean(T\_minus\_j)  
J\_T = groups\*b2 - (groups-1)\*T\_bar\_dot  
  
jk\_var = (groups-1)/groups \* sum((T\_minus\_j - T\_bar\_dot)^2)  
paste("Values for k=",k,"; b2:",round(b2,4),", J\_T:",round(J\_T,6),", T\_bar\_dot:",round(T\_bar\_dot,4),"SD", round(sqrt(jk\_var),4))

## [1] "Values for k= 1 ; b2: 0.0267 , J\_T: -0.00102 , T\_bar\_dot: 0.027 SD 0.0067"

### Part C

for(iter in 1:10){  
 k=1  
 norm\_rand\_1 = as.data.frame(rnorm(100,1,sqrt(2)))  
 colnames(norm\_rand\_1) = c("jackknife")  
 groups = length(norm\_rand\_1$jackknife)/k  
 jack\_group = rep(1:groups,each =k)  
 jackknife= cbind.data.frame(norm\_rand\_1,jack\_group)  
 jack\_mean = mean(norm\_rand\_1$jackknife)  
 b2 = sum((norm\_rand\_1$jackknife -jack\_mean)^4)/sum((norm\_rand\_1$jackknife -jack\_mean)^2)^2  
   
 T\_minus\_j = NULL  
   
 for(i in 1:groups){  
 jk\_minus\_j = jackknife$jackknife[jackknife$jack\_group != i]  
 jk\_minus\_j\_mean = mean(jk\_minus\_j)  
 T\_minus\_j[i] = sum((jk\_minus\_j -jk\_minus\_j\_mean)^4)/sum((jk\_minus\_j -jk\_minus\_j\_mean)^2)^2  
 }  
   
 T\_bar\_dot = mean(T\_minus\_j)  
 J\_T = groups\*b2 - (groups-1)\*T\_bar\_dot  
   
 jk\_var = (groups-1)/groups \* sum((T\_minus\_j - T\_bar\_dot)^2)  
   
 print(paste("Values for k=",k,"; b2:",round(b2,4),", J\_T:",round(J\_T,6),", T\_bar\_dot:",round(T\_bar\_dot,4),"SD", round(sqrt(jk\_var),4)))  
}

## [1] "Values for k= 1 ; b2: 0.0325 , J\_T: 0.000518 , T\_bar\_dot: 0.0328 SD 0.0031"  
## [1] "Values for k= 1 ; b2: 0.029 , J\_T: 0.000864 , T\_bar\_dot: 0.0293 SD 0.0053"  
## [1] "Values for k= 1 ; b2: 0.0242 , J\_T: 0.000204 , T\_bar\_dot: 0.0244 SD 0.0028"  
## [1] "Values for k= 1 ; b2: 0.0255 , J\_T: -8.5e-05 , T\_bar\_dot: 0.0257 SD 0.0026"  
## [1] "Values for k= 1 ; b2: 0.0273 , J\_T: 5.7e-05 , T\_bar\_dot: 0.0275 SD 0.0027"  
## [1] "Values for k= 1 ; b2: 0.031 , J\_T: 0.000543 , T\_bar\_dot: 0.0313 SD 0.0041"  
## [1] "Values for k= 1 ; b2: 0.0232 , J\_T: 3.8e-05 , T\_bar\_dot: 0.0235 SD 0.0022"  
## [1] "Values for k= 1 ; b2: 0.032 , J\_T: 0.000469 , T\_bar\_dot: 0.0323 SD 0.0038"  
## [1] "Values for k= 1 ; b2: 0.0265 , J\_T: 0.000228 , T\_bar\_dot: 0.0268 SD 0.0028"  
## [1] "Values for k= 1 ; b2: 0.0262 , J\_T: 0.000147 , T\_bar\_dot: 0.0265 SD 0.0024"

It seems that T\_bar\_dot is always close to b2, but always slightly higher. J\_T is usually close to 0, as well as the standard deviation.