hw6

Jared Yu

June 7, 2018

```r
# Question 1

library(xtable)
#1
#a)
data = read.table("q1_data.txt")
LSAT = t(data[1,])
GPA = t(data[2,])
V1 = var(LSAT)
V2 = var(GPA)
Cov = cov(LSAT,GPA)
Corr = Cov/sqrt(V1*V2)

#b)
jackknife = c(rep(0,ncol(data)))
for(i in 1:ncol(data)){
    L = t(data[1,-i])
    G = t(data[2,-i])
    v1 = var(L)
    v2 = var(G)
    c = cov(L,G)
    jackknife[i] = c/sqrt(v1*v2)
}
n=15
se = sqrt((n-1)*sum((jackknife-Corr)^2)/n)

B = 200
bootstrap = c(rep(0,B))
for(i in 1:B){
    new = sample(data,size=ncol(data),replace=T)
    L = t(new[1,])
    G = t(new[2,])
    v1 = var(L)
    v2 = var(G)
    c = cov(L,G)
    bootstrap[i] = c/sqrt(v1*v2)
}
se = sd(bootstrap)

#c)
B = 1000
bootstrap = c(rep(0,B))
for(i in 1:B){
    new = sample(data,size=ncol(data),replace=T)
    L = t(new[1,])
    G = t(new[2,])
    v1 = var(L)
    v2 = var(G)
```

```
    c = cov(L,G)
    bootstrap[i] = c/sqrt(v1*v2)
}
se = sd(bootstrap)
CI.l=Corr-1.96*se
CI.u=Corr+1.96*se

B = 2000
bootstrap = c(rep(0,B))
z = c(rep(0,B))
for(i in 1:B){
    new = sample(data,size=ncol(data),replace=T)
    L = t(new[1,])
    G = t(new[2,])
    v1 = var(L)
    v2 = var(G)
    c = cov(L,G)
    bootstrap[i] = c/sqrt(v1*v2)

    k=200
    bt = c(rep(0,k))
    ind <- matrix(sample(1:ncol(data),size=ncol(data)*k,replace=T),nr=k,nc=n)
    bt <- sapply(1:k,function(x){cor(t(new[ind[x,]][1,]),t(new[ind[x,]][2,]))})

    s = sd(bt)
    z[i]=(bootstrap[i]-Corr)/s
}
se = sd(bootstrap)
CI.l=Corr+z[order(z)][5]*se
CI.u=Corr+z[order(z)][95]*se

# Question 2

set.seed(2018)
n = 50
X=runif(n,0,3)


#c,d,e)
theta=max(X)
B=5000
bootstrap_P=c(rep(0,B))
for(i in 1:B){
    boot = runif(length(X),0,theta)
    bootstrap_P[i]=max(boot)
}
var(bootstrap_P) #0.00317

bootstrap_NP = c(rep(0,B))
```

```
for(i in 1:B){
    boot = sample(X,size=length(X),replace=T)
    bootstrap_NP[i]=max(boot)
}
var(bootstrap_NP) #0.00115

# true var
50*9/(52*51^2)  # 0.003327123

y=seq(0,3,length=1000)
d=function(x){50*(x/3)^49}
f=sapply(y,d)

par(mfrow=c(1,2))
hist(bootstrap_P,freq=F,main="Histogram of theta using parametric bootstrapping.")
lines(y,f)
hist(bootstrap_NP,freq=F,main="Histogram of theta using non-parametric bootstrappin
g.")
lines(y,f)

# Question 3

source("hw2_modifiedGA.r")

## test functions
tf1 = function(x){
    t = c(0.1, 0.13, 0.15, 0.23, 0.25, 0.4, 0.44, 0.65, 0.76, 0.78, 0.81)
    h = c(4, -5, 3, -4, 5, -4.2, 2.1, 4.3, -3.1, 2.1, -4.2)
    temp = 0
    for(i in 1:11) {
        temp = temp + h[i]/2 * (1 + sign(x - t[i]))
    }
    return(temp)
}
tf2 = function(x){
    return((4*x-2)+2*exp(-16*(4*x-2)^2))
}
set.seed(2435)
n = 512
x = (0:(n-1))/n
tf1.sigma2 = integrate(function(x){(tf1(x))^2},0,1,subdivisions=10000)$value/5^2
tf2.sigma2 = integrate(function(x){(tf2(x))^2},0,1,subdivisions=10000)$value/5^2
y1 = tf1(x) + rnorm(n)*sqrt(tf1.sigma2)
y2 = tf2(x) + rnorm(n)*sqrt(tf2.sigma2)

png('plot2.png',width=980,height=480)
par(mfrow=c(1,2))
plot(x,y1,main="test function 1",ylab='y',cex=0.5)
lines(x,tf1(x))
```

```
plot(x,y2,main="test function 2",ylab='y',cex=0.5)
lines(x,tf2(x))
dev.off()
fit1 = ga.piecewise(x=x,y=y1,MDL)
fit2 = ga.piecewise(x=x,y=y2,MDL)

png('plot3.png',width=980,height=480)
par(mfrow=c(1,2))
plot(x,y1,main="a regression curve estimates for \n test function 1",cex=0.5,ylab='y')
lines(x,tf1(x),lwd=5,col='grey')
lines(x,fit1$f)
par(new=TRUE)
plot(x,y1,main="a regression curve estimates for \n test function 1",cex=0.5,ylab='y')
plot(x,y2,main="a regression curve estimates for \n test function 2",cex=0.5,ylab='y')
lines(x,tf2(x),lwd=5,col='grey')
lines(x,fit2$f)
par(new=TRUE)
plot(x,y2,main="a regression curve estimates for \n test function 2",cex=0.5,ylab='y')
dev.off()

set.seed(24353)
pair1 = bst.pair(x,y1,200)
set.seed(24354)
pair2 = bst.pair(x,y2,200)
set.seed(24355)
res1 = bst.res(x,y1,200,fit1$f)
set.seed(24356)
res2 = bst.res(x,y2,200,fit2$f)

save(pair1, pair2, file = "pair.RData")
save(res1, res2, file = "res.RData")

CI_pair1 = getCI(pair1.fit)
CI_pair2 = getCI(pair2.fit)
CI_res1 = getCI(res1$fit)
CI_res2 = getCI(res2$fit)
pair1.fit = getpair(pair1)
pair2.fit = getpair(pair2)

png('plot4.png',width=980,height=980)
par(mfrow = c(2,2))
plot(x,y1,main="test function 1 with bootstrapping pairs",cex=0.5,ylab='y')
lines(x,CI_pair1[2,],col="grey",lty=1,lwd=2)
lines(x,CI_pair1[1,],col="grey",lty=1,lwd=2)
polygon(c(x, rev(x)), c(CI_pair1[2,], rev(CI_pair1[1,])), col = "grey", border = NA)
par(new=TRUE)
plot(x,y1,main="test function 1 with bootstrapping pairs",cex=0.5,ylab='y')
lines(x,tf1(x))
plot(x,y1,main="test function 1 with bootstrapping residuals",cex=0.5,ylab='y')
```

```
lines(x,CI_res1[2,],col="grey",lty=1,lwd=2)
lines(x,CI_res1[1,],col="grey",lty=1,lwd=2)
polygon(c(x, rev(x)), c(CI_res1[2,], rev(CI_res1[1,])), col = "grey", border = NA)
par(new=TRUE)
plot(x,y1,main="test function 1 with bootstrapping residuals",cex=0.5,ylab='y')
lines(x,tf1(x))
plot(x,y2,main="test function 2 with bootstrapping pairs",cex=0.5,ylab='y')
lines(x,CI_pair2[2,],col="grey",lty=1,lwd=2)
lines(x,CI_pair2[1,],col="grey",lty=1,lwd=2)
polygon(c(x, rev(x)), c(CI_pair2[2,], rev(CI_pair2[1,])), col = "grey", border = NA)
par(new=TRUE)
plot(x,y2,main="test function 2 with bootstrapping pairs",cex=0.5,ylab='y')
lines(x,tf2(x))
plot(x,y2,main="test function 2 with bootstrapping residuals",cex=0.5,ylab='y')
lines(x,CI_res2[2,],col="grey",lty=1,lwd=2)
lines(x,CI_res2[1,],col="grey",lty=1,lwd=2)
polygon(c(x, rev(x)), c(CI_res2[2,], rev(CI_res2[1,])), col = "grey", border = NA)
par(new=TRUE)
plot(x,y2,main="test function 2 with bootstrapping residuals",cex=0.5,ylab='y')
lines(x,tf2(x))
dev.off()

# Question 4

alpha = 0.05
n = 50
B = 1000
set.seed(100)
sample = rexp(n*B)
z = qnorm(1-alpha/2)
asyCI = matrix(0, nrow=1000, ncol=2)
exaCI = matrix(0, nrow=1000, ncol=2)
for (i in 1:B){
    smallsample = sample[(n*(i-1)):(n*(i))]
    lambdahat = n/sum(smallsample)
    asyCI[i,1] = lambdahat*exp(-z/sqrt(n))
    asyCI[i,2] = lambdahat*exp(z/sqrt(n))
    exaCI[i,1] = lambdahat*qgamma(alpha/2,n,1)/n
    exaCI[i,2] = lambdahat*qgamma(1-alpha/2,n,1)/n
}
sum((asyCI[,2] > 1) & (asyCI[,1] < 1))/B
sum((exaCI[,2] > 1) & (exaCI[,1] < 1))/B
mean(asyCI[,2]-asyCI[,1])
mean(exaCI[,2]-exaCI[,1])
```