HW7_GUO

QING GUO

10/21/2019

#Problem 2

##Part a

```
#install.packages("tinytex")
library(tinytex)
#tinytex::install_tinytex()
library(quantmod)
#1)fetch data from Yahoo
#AAPL prices
apple08 <- getSymbols('AAPL', auto.assign = FALSE, from = '2008-1-1', to =
                        "2008-12-31")[,6]
#market proxy
rm08<-getSymbols('^ixic', auto.assign = FALSE, from = '2008-1-1', to =
                   "2008-12-31")[,6]

#log returns of AAPL and market
logapple08<- na.omit(ROC(apple08)*100)
logrm08<-na.omit(ROC(rm08)*100)

#OLS for beta estimation
beta_AAPL_08<-summary(lm(logapple08~logrm08))$coefficients[2,1]

df08<-cbind(logapple08,logrm08)
set.seed(666)
Boot_times=1000
sd.boot=rep(0,Boot_times)
for(i in 1:Boot_times){
  # nonparametric bootstrap
  bootdata=df08[sample(nrow(df08), size = 251, replace = TRUE),]
  sd.boot[i]= coef(summary(lm(AAPL.Adjusted~IXIC.Adjusted, data = bootdata)))[2,2]
}
```

##Part b

```
library(knitr)
time1<-system.time({
url<-"https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat"
a<-read.table(url,header=T,sep=" ",skip=1,fill=T)
for (i in 1:30){
  if (is.na(a[i,6])==TRUE){
    a[i,2:6]=a[i,1:5]
    a[i,1]=floor((i+2)/3)
  }
}
bb<-matrix(1:2,1,2)
for (i in 2:6){
```

```
  b<-cbind(a[,1],a[,i])
  bb<-rbind(bb,b)
}
data<-as.data.frame(bb[2:151,] )
set.seed(666)
Boot=100
bs=function(i){
  index<-c(sample(1:30,size=30,replace=TRUE),
           sample(31:60,size=30,replace=TRUE),
           sample(61:90,size=30,replace=TRUE),
           sample(31:60,size=30,replace=TRUE),
           sample(91:120,size=30,replace=TRUE),
           sample(121:150,size=30,replace=TRUE))
  sensory<-data[index,]
  coef(lm(V2 ~ V1, data=sensory))
}
N = 100
coef<-sapply(1:N, bs)
})
```

##Part c

```
library("parallel")
time2<-system.time({
  c1<-makeCluster(8)
  url<-"https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat"
  a<-read.table(url,header=T,sep=" ",skip=1,fill=T)
  for (i in 1:30){
    if (is.na(a[i,6])==TRUE){
      a[i,2:6]=a[i,1:5]
      a[i,1]=floor((i+2)/3)
    }
  }
  bb<-matrix(1:2,1,2)
  for (i in 2:6){
    b<-cbind(a[,1],a[,i])
    bb<-rbind(bb,b)
  }
  data<-as.data.frame(bb[2:151,] )
  set.seed(666)
  Boot<-100
  bs<-function(i){
    index<-c(sample(1:30,size=30,replace=TRUE),
             sample(31:60,size=30,replace=TRUE),
             sample(61:90,size=30,replace=TRUE),
             sample(31:60,size=30,replace=TRUE),
             sample(91:120,size=30,replace=TRUE),
             sample(121:150,size=30,replace=TRUE))
    sensory<-data[index,]
    coef(lm(V2 ~ V1, data=sensory))
  }
  coef<-sapply(1:100,bs)
  clusterExport(c1, "data")
```

```
  coef<-parSapply(c1, 1:100, bs)
  stopCluster(c1)
})
kable(rbind(time1,time2))
```

	user.self	sys.self	elapsed	user.child	sys.child
time1	0.148	0.016	0.340	0.000	0.000
time2	0.149	0.026	2.582	0.001	0.014

#Problem 3 ##Part a

```
time1<-system.time({
  a<-function(i){
    df<-function(x){
      df<-(3^x)*log(3)-cos(x)-5*sin(5*x)
      return(df)
    }
    f<-function(x){
      f<-3^x+sin(x)-5*cos(5*x)
      return(f)
    }
    e<-c()
    d<-abs(f(0))
    x<-0
    while (d>0.05){
      x<-x-(f(x)/df(x))
      d<-abs(f(x))
      e<-c(e,x)
    }
  }
  b<-sapply(1:1000,a)
})
```

##Part b

```
time2<-system.time({
  c1<-makeCluster(8)
  a<-function(i){
    df<-function(x){
      df<-(3^x)*log(3)-cos(x)-5*sin(5*x)
      return(df)
    }
    f<-function(x){
      f<-3^x+sin(x)-5*cos(5*x)
      return(f)
    }
    e<-c()
    d<-abs(f(0))
    x<-0
    while (d>0.05){
      x<-x-(f(x)/df(x))
```

```
      d<-abs(f(x))
      e<-c(e,x)
    }
  }
  b<-parSapply(c1,1:1000,a)
  stopCluster(c1)
})
kable(rbind(time1,time2))
```

	user.self	sys.self	elapsed	user.child	sys.child
time1	1.291	0.026	1.317	0.000	0.000
time2	0.006	0.024	2.368	0.003	0.014

#Problem 4

Part a

No. It is not an unbiased estimate and also may not have minimum variance.

Part c

```
    set.seed(1256)
    theta <- as.matrix(c(1,2),nrow=2)
    X <- cbind(1,rep(1:10,10))
    h <- X%*%theta+rnorm(100,0,0.2)
  #quick gradient descent
  #need to make guesses for both Theta0 and Theta1, might as well be close
f<-function(start){
  alpha <- 0.0000001 # this is the step size
  m <- 100 # this is the size of h
  tolerance <- 0.000000001 # stopping tolerance
  theta0 <- c(start,rep(0,999)) # I want to try a guess at 1, setting up container for max 1000 iters
  theta1 <- c(start,rep(0,999))
  i <- 2 #iterator, 1 is my guess (R style indecies)
  #current theta is last guess
  current_theta <- as.matrix(c(theta0[i-1],theta1[i-1]),nrow=2)
  #update guess using gradient
  theta0[i] <-theta0[i-1] - (alpha/m) * sum(X %*% current_theta - h)
  theta1[i] <-theta1[i-1] - (alpha/m) * sum((X %*% current_theta - h)*rowSums(X))
  rs_X <- rowSums(X) # can precalc to save some time
  z <- 0
  while(abs(theta0[i]-theta0[i-1])>tolerance && abs(theta1[i]-theta1[i-1])>tolerance && z<2000000){
    if(i==1000){theta0[1]=theta0[i]; theta1[1]=theta1[i]; i=1; } ##cat("z=",z,"\n",sep="")}
    z <- z + 1
    i <- i + 1
    current_theta <- as.matrix(c(theta0[i-1],theta1[i-1]),nrow=2)
    theta0[i] <-theta0[i-1] - (alpha/m) * sum(X %*% current_theta - h)
    theta1[i] <-theta1[i-1] - (alpha/m) * sum((X %*% current_theta - h)*rs_X)
```

```
  }
}
system.time({
  c1<-makeCluster(8)
  clusterExport(c1, "X")
  clusterExport(c1, "h")
  result<-parSapply(c1, 1:10, f)
  stopCluster(c1)
})
```

```
##    user  system elapsed
##   0.012   0.036  38.157
```