my.smooth.forKS <- function(data,xindex,yindex,ind.sqrt=T){

#X and Y variable for data[[xindex]] and data[[yindex]]

X <- data[[xindex]]

Y <- data[[yindex]]

#set up two conditions for sqrt or not sqrt

if(ind.sqrt){

#smooth.spline based on X and sqrt(Y)

smsp.strcv <- smooth.spline(X,sqrt(Y))

#get the residual from sqrt(Y) and approx(sqrt(Y))

smspcv.resid <- sqrt(Y)-approx(smsp.strcv$x, smsp.strcv$y, X)$y

} else {

#smooth.spline based on X and Y

smsp.strcv <- smooth.spline(X,Y)

#get the residual from Y and approx(Y)

smspcv.resid <- Y-approx(smsp.strcv$x, smsp.strcv$y, X)$y

}

#get the standard deviation of the residual

sd.resid <- sqrt(sum(smspcv.resid^2)/(nrow(data)-smsp.strcv$df))

###stud.resid <- smspcv.resid/sd.resid

###D <- ks.test(stud.resid,pnorm)$statistic

#set the approx x and y to my.smooth

my.smooth <- approx(smsp.strcv$x, smsp.strcv$y, X)$y

###list(D = D, raw.resid=smspcv.resid,sd.resid = sd.resid,smooth=my.smooth)

list(raw.resid=smspcv.resid, sd.resid = sd.resid,smooth=my.smooth)

}

my.boot.smooth <- function(data=NOAA,xindex=3,yindex=2,nboot=1000,confidence=0.95){

#set original info

par(mfrow=c(1,1))

smooth.dist <- NULL

#set str1 as the base smooth from the data, xindex, yindex

str1 <- my.smooth.forKS(data,xindex,yindex)

#set mysmoth as base smooth

mysmooth <- str1$smooth

#set mysd as base sd of residual

mysd <- str1$sd.resid

#set the list of residual data to myrsd

myrsd <- str1$raw.resid

#in this example n is 37

n <- length(mysmooth)

#set mybootdata as a copy of the data which will use in the for loop

mybootdata <- data

#start for loop for nboot time

for(i in 1:nboot){

#set bres as the random sample of myrsd, size 37

bres <- sample(myrsd,length(myrsd),replace=T)

#add bres on mysmooth, named boot.dat

boot.dat <- (mysmooth+bres)

###print(boot.dat)

#set boot.dat to mybootdata[[yindex]] as a new dataset

mybootdata[[yindex]]<-boot.dat

#get the smooth again based on new dataset and named bstr

bstr <- my.smooth.forKS(mybootdata,xindex,yindex,F)

boot.smooth <- bstr$smooth

#smooth.dist: get the distance from (boot.smooth-mysmooth), size: 1 list of 37 elements, repeat nboot times (nboot, 37)

smooth.dist<-rbind(smooth.dist,boot.smooth-mysmooth)

}

#get the col number from smooth.dist

n<-length(smooth.dist[1,])

#get value of alpha

alpha<-1-confidence

#initial Lower bound and upper bound

LB<-NULL

UB<-NULL

for(i in 1:n){

#s1 is the sorting result of smooth.dist of one paticular column, ascending

s1<-sort(smooth.dist[,i])

#length of s1

n2<-length(s1)

v1<-c(1:n2)/n2

#get the lower bound and upper bound on each distance

bvec<-approx(v1,s1,c(alpha/2,1-alpha/2))$y

#get the lower bound and upper bound set on mysmooth

LB<-c(LB,mysmooth[i]-bvec[2])

UB<-c(UB,mysmooth[i]-bvec[1])

}

#plot the data based on xindex and the largest of four ranges, don't show the linear plot on the picture

plot(rep(data[[xindex]],4),c(LB,mysmooth,UB,sqrt(data[[yindex]])),xlab="X",ylab="Y",type="n")

#add scatter point on blank picture

points(data[[xindex]],sqrt(data[[yindex]]))

#set the order of x to o1

o1<-order(data[[xindex]])

#skech the smooth line and lower and upper bound in the picture.

lines(data[[xindex]][o1],LB[o1],col=2)

lines(data[[xindex]][o1],UB[o1],col=2)

lines(data[[xindex]][o1],mysmooth[o1],col=3)

}

my.boot.smooth()

