Question 1

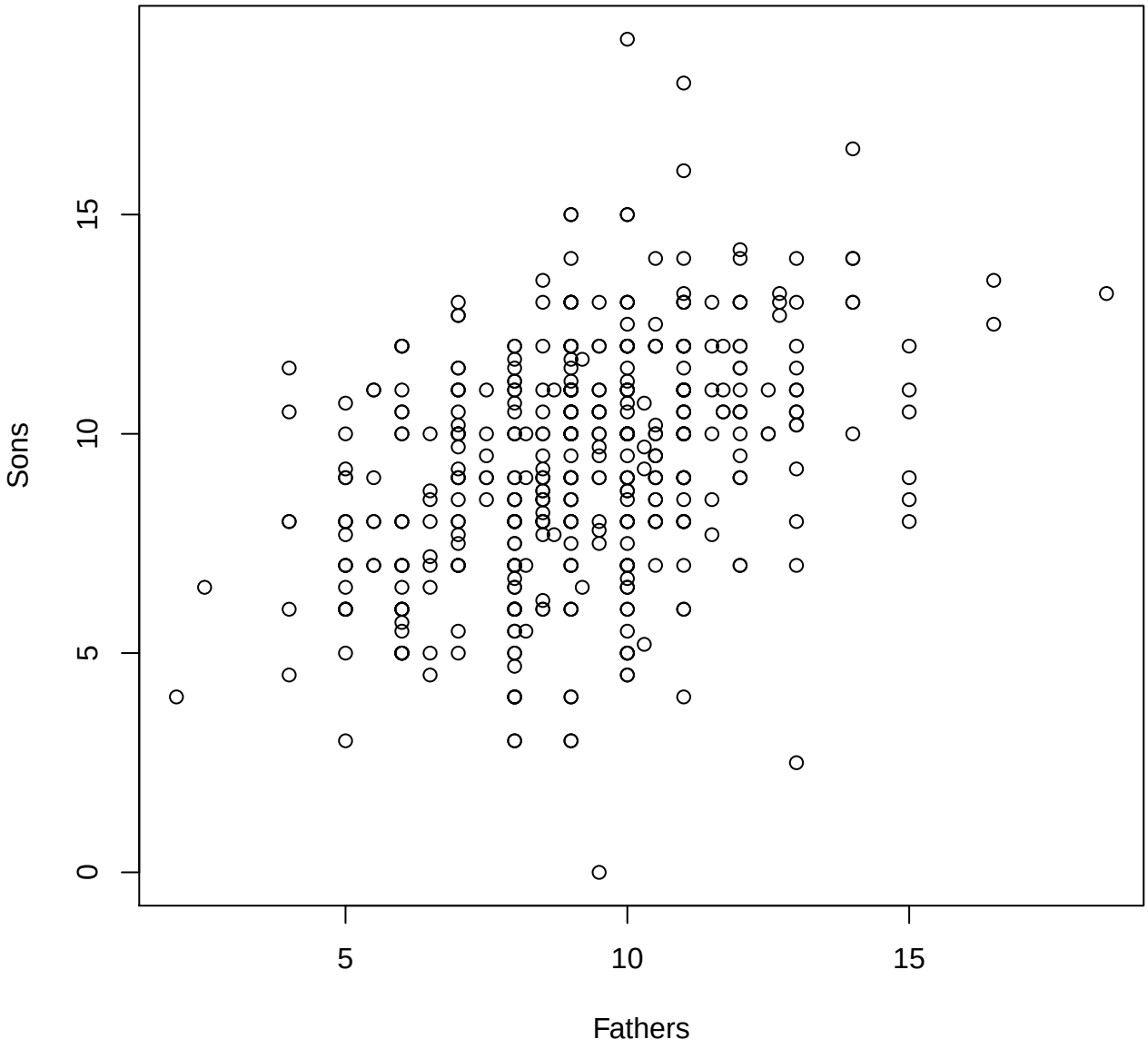
a

```
data <- read.table( "Galton.dat" )

sons <- data[data$V4==1,]
daughters <- data[data$V4==0,]
fathers <- unique( data[,1:2])


summary(sons$V5)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   7.500   9.200   9.234  11.000  19.000

sd( sons$V5)

## [1] 2.623688

max( sons$V5 ) - min( sons$V5 )

## [1] 19

IQR( sons$V5 )

## [1] 3.5

summary( daughters$V5 )

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -4.000   2.500   4.000   4.103   5.500  10.500

sd( daughters$V5)

## [1] 2.356053

max( daughters$V5 ) - min( daughters$V5 )

## [1] 14.5

IQR( daughters$V5 )

## [1] 3
```

Both fathers and mothers that have large families would be over-represented by the data, since there is one row per offspring. Parents of families that had more than one son or daughter would appear more than once, leading to them being mis-represented. This problem can be rectified with the unique function that was applied to the variable *fathers.*

b

```
plot( x=sons$V2,y=sons$V5, xlab="Fathers", ylab="Sons")
```

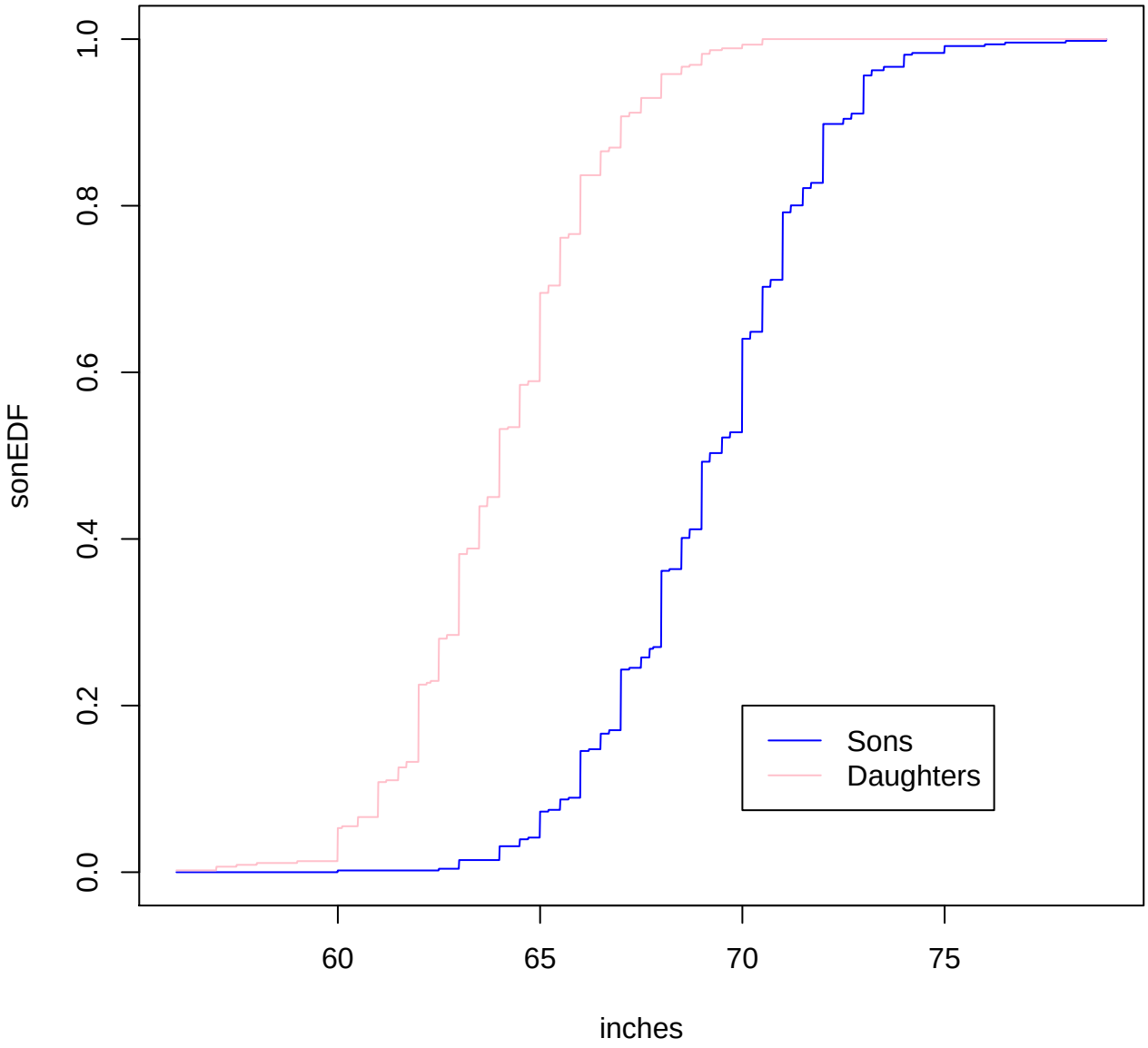


Question 2

a

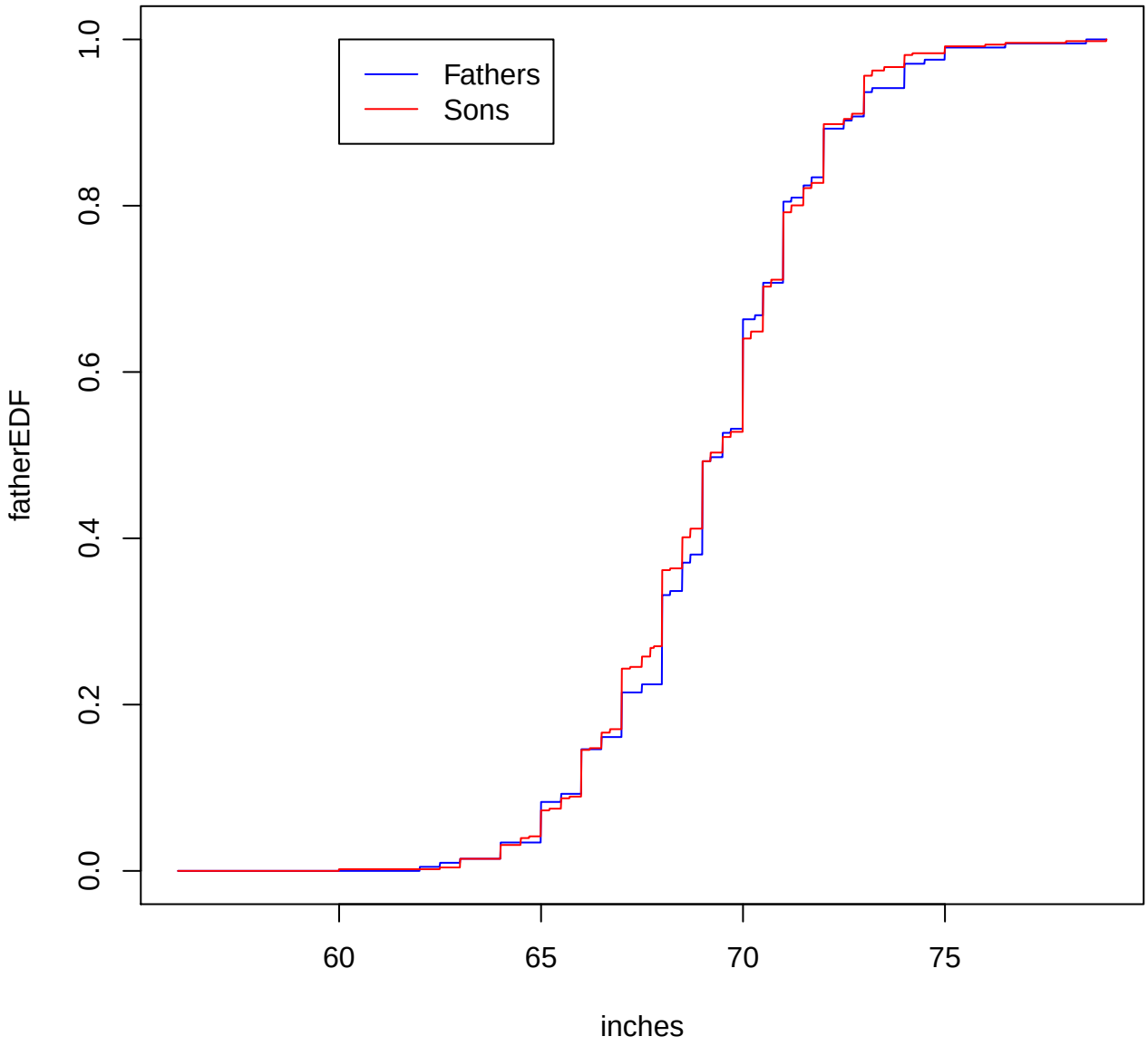
```
GenerateEDF <- function( depth, frame, start, end ){
    EDF <- numeric( (end-start)*depth )
    i <- 0
    for( i in 1:length( EDF )){
        EDF[i] <- length( frame[frame<=(start+(i/depth))] ) / length( frame )
    }
    EDF
}
depth <-  100

start <- min( sons$V5, daughters$V5 )
end <- max( sons$V5, daughters$V5 )
```

```
sonEDF <- GenerateEDF( depth, sons$V5, start, end )
daughterEDF <- GenerateEDF( depth, daughters$V5, start, end )

inches <-  numeric( (end-start)*depth )
for( i in 1:length(inches)){
    inches[i] <- start + i/depth + 60
}

plot( x=inches, y=sonEDF, type="l", col="blue" )
lines( x=inches,y=daughterEDF, col="pink")
legend( 70,.2, legend=c("Sons","Daughters"), lty = c(1,1) , col=c("blue","pink"))
```



b

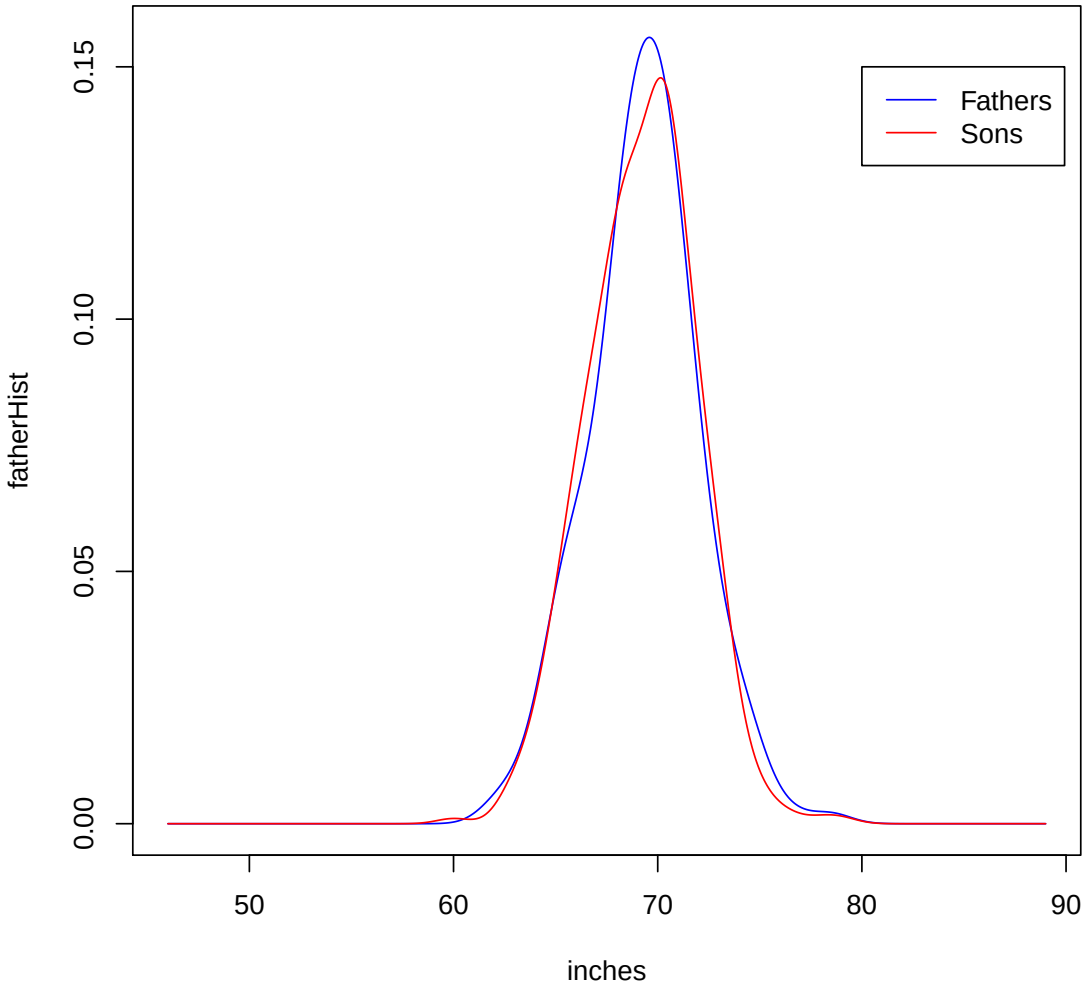
```
fatherEDF <- GenerateEDF( depth, fathers$V2, start, end )

plot( x = inches, y=fatherEDF, type="l", col="blue" )
lines(x=inches,y=sonEDF,col="red")
legend( 60,1, legend=c("Fathers","Sons"), lty = c(1,1) , col=c("blue","red"))
```



Question 3.

```
depth <-  100


NormalKernal <- function( u ){
    (1/sqrt( 2*pi))*exp( (-1*u^2)/2.0 )
}
```

```
GenerateKernalSmoothHistogram <- function( depth, frame, start, end, kernal ){
    kSmooth <- numeric( (end-start)*depth )
    i <- 0
    h <- 1.06*sd(frame)*as.numeric(length(frame))^(-1.0/5.0)
    for( i in 1:length(kSmooth) ){
        kSmooth[i] <- 0
        y <- (start+(i/depth))
        for( j in 1:length(frame)){
            kSmooth[i] <- kSmooth[i] + kernal( (y - frame[j])/h )
        }
        kSmooth[i] <- kSmooth[i] / (length(frame)*h)
    }
    kSmooth
}

start <- min( sons$V5, daughters$V5 ) - 10
end <- max( sons$V5, daughters$V5 ) + 10

sonHist <- GenerateKernalSmoothHistogram( depth, sons$V5, start, end, NormalKernal )
daughterHist <- GenerateKernalSmoothHistogram( depth, daughters$V5, start, end, NormalKernal )
fatherHist <- GenerateKernalSmoothHistogram( depth, fathers$V2, start, end, NormalKernal )
inches <-  numeric( (end-start)*depth )
for( i in 1:length(inches)){
    inches[i] <- start + i/depth + 60
}

plot( x=inches,y=fatherHist, type="l", col="blue" )
lines(x=inches,y=sonHist, col="red" )
legend( 80,.15, legend=c("Fathers","Sons"), lty = c(1,1) , col=c("blue","red"))
```


Fathers
Sons
fatherHist
0.00
0.05
0.10
0.15
50
60
70
80
90
inches