FE 582 Assignment 2

Mugdha

10/14/2020

Extracting data for 2010,2011,2012

```
library(XML)
years = 2010:2012
womenURLs =
  c("results/2010/2010cucb10m-f.htm",
    "results/2011/2011cucb10m-f.htm",
    "results/2012/2012cucb10m-f.htm")

# Retrieve data from web site, find pre formatted text,
# return as a character vector.
extractTable = function(url, year, file = NULL){
    ubase = "http://www.cherryblossom.org/"
    url = paste(ubase, url, sep = "")
    doc = htmlParse(url)
    preNode = getNodeSet(doc, "//pre")
    txt = xmlValue(preNode[[1]])
    els = strsplit(txt, "\\r\\n")[[1]]



  if (is.null(file)) return(els)

  elss = writeLines(els, con = file)
  return(elss)
}
womenTables = mapply(extractTable, url = womenURLs, year = years)
names(womenTables) = years
save(womenTables, file = "CBWomenTextTables.rda")
```

Finding the starting and ending positions of the columns

```
findColLocs = function(spacerRow) {

  spaceLocs = gregexpr(" ", spacerRow)[[1]]
  rowLength = nchar(spacerRow)

  if (substring(spacerRow, rowLength, rowLength) != " ")
    return( c(0, spaceLocs, rowLength + 1))
  else return(c(0, spaceLocs))
}
```

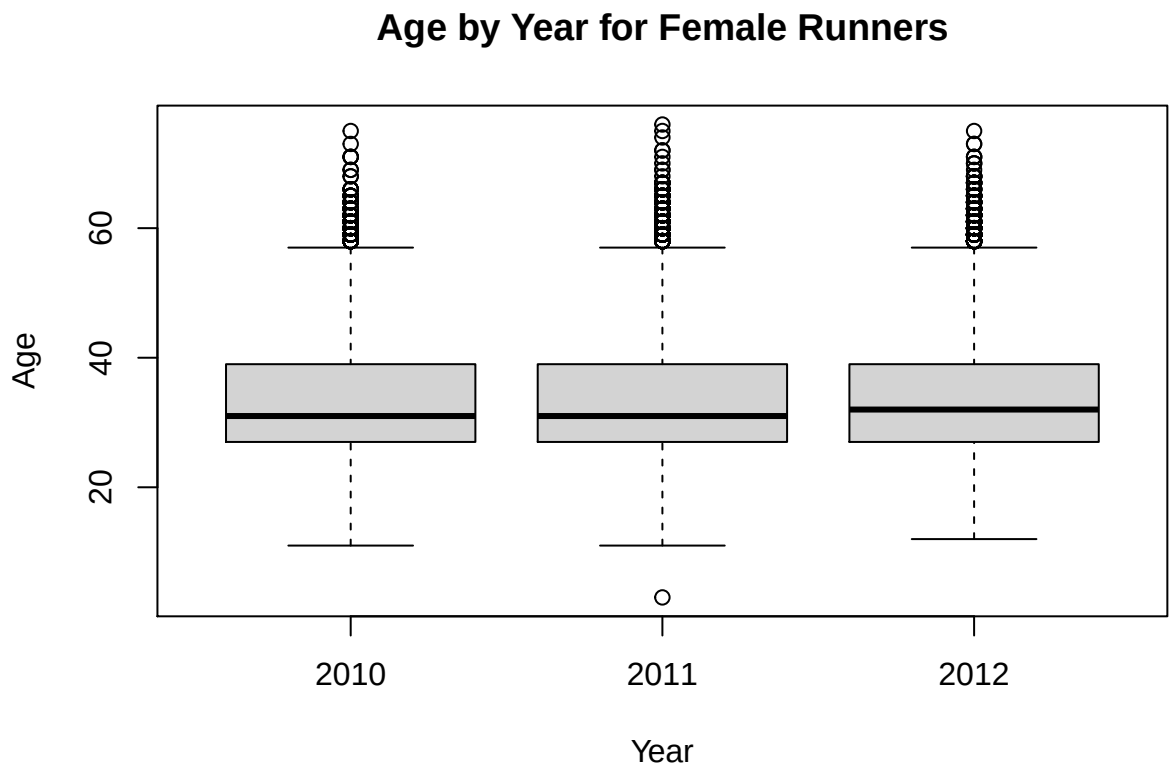
```
#We encapsulate into a function the code to extract the locations of the desired columns. We need, as i
selectCols = function(colNames, headerRow, searchLocs)
{
  sapply(colNames,
         function(name, headerRow, searchLocs)
         {
           startPos = regexpr(name, headerRow)[[1]]
           if (startPos == -1)
             return( c(NA, NA) )

           index = sum(startPos >= searchLocs)
           c(searchLocs[index] + 1, searchLocs[index + 1] - 1)
         },
         headerRow = headerRow, searchLocs = searchLocs )
}


extractVariables =
  function(file, varNames =c("name", "home", "ag", "gun",
                             "net", "time")){

    ## Extract all variables corresponding with the right columns
    ## Find the index of the row with =s
    # Find which row the space located

    eqIndex = grep("^===", file)

    # Extract the two key rows and the data
    # The one row before the space row is header row
    # The rows after the space row is body
    spacerRow = file[eqIndex]
    headerRow = tolower(file[ eqIndex - 1 ])
    body = file[ -(1 : eqIndex) ]

    # Remove footnotes and blank rows
    footnotes = grep("^[[:blank:]]*(\\*|\\#)", body)
    if ( length(footnotes) > 0 ) body = body[ -footnotes ]
    blanks = grep("^[[:blank:]]*$", body)
    if (length(blanks) > 0 ) body = body[ -blanks ]

    # Obtain the starting and ending positions of variables
    searchLocs = findColLocs(spacerRow)
    locCols = selectCols(varNames, headerRow, searchLocs)

    Values = mapply(substr, list(body), start = locCols[1, ],
                    stop = locCols[2, ])
    colnames(Values) = varNames

    invisible(Values)
  }

womenMat = lapply(womenTables, extractVariables)
save(womenMat, file = "cbWomenTables.rda")
```

```
convertTime = function(time) {

  ## Convert Time format from char to number by mins

  timePieces = strsplit(time, ":")
  timePieces = sapply(timePieces, as.numeric)
  sapply(timePieces, function(x) {
    if (length(x) == 2) x[1] + x[2]/60
    else 60*x[1] + x[2] + x[3]/60
  })
}

createDF = function(Res, year, sex)
{

  ## Creat dataframe, determine which time to use,
  ## Remove the rows with missing time

  # Determine which time to use
  if ( !is.na(Res[1, 'net']) ) useTime = Res[ , 'net']
  else if ( !is.na(Res[1, 'gun']) ) useTime = Res[ , 'gun']
  else useTime = Res[ , 'time']

  # Remove # and * and blanks from time
  useTime = gsub("[#\\*[:blank:]]", "", useTime)
  runTime = convertTime(useTime[ useTime != "" ])

  # Drop rows with no time
  # Drop rows with no age
  Res = Res[ useTime != "", ]

  Results = data.frame(year = rep(year, nrow(Res)),
                       sex = rep(sex, nrow(Res)),
                       name = Res[ , 'name'], home = Res[ , 'home'],
                       age = suppressWarnings(as.numeric(Res[, 'ag'])),
                       runTime = runTime,
                       stringsAsFactors = FALSE)

  invisible(Results)
}
womenMat = sapply(womenTables, extractVariables)

womenDF = mapply(createDF, womenMat, year = years,
                 sex = rep("F", 12), SIMPLIFY = FALSE)



cbWomen = do.call(rbind, womenDF)
save(cbWomen, file = "cbWomen.rda")
```

Box Plot of Age by Year for Female Runners

Solution:

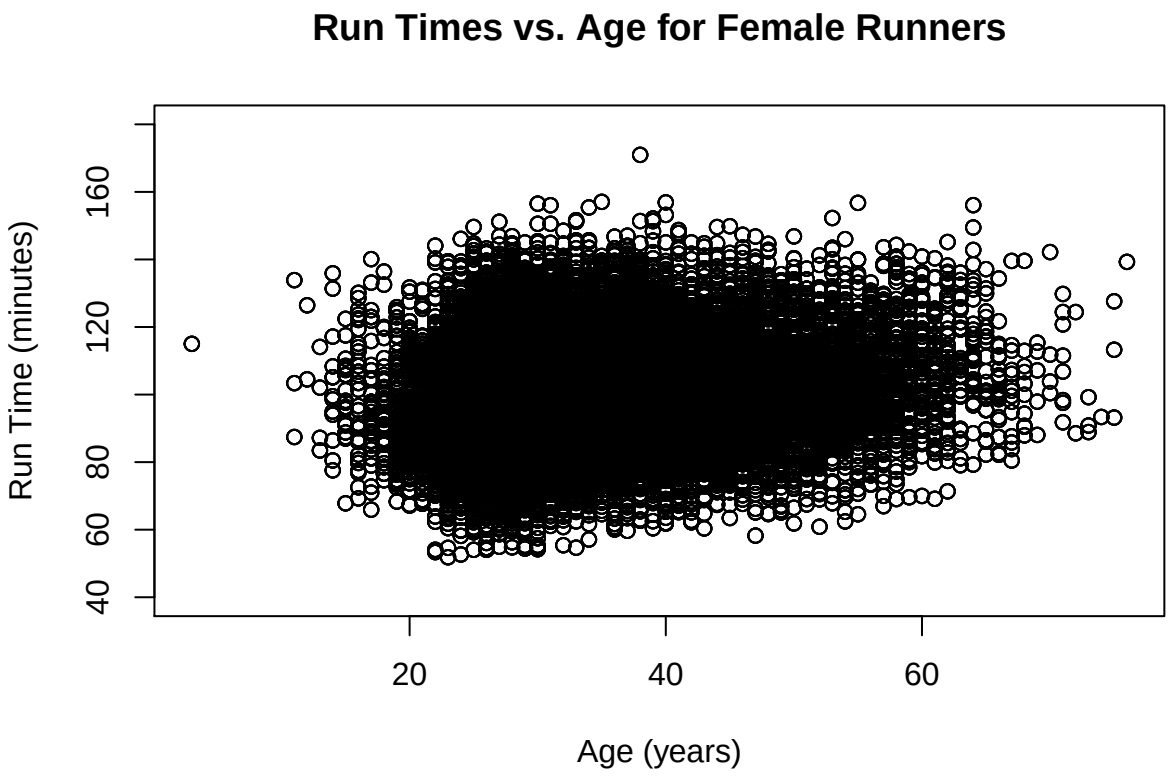
```
library(RColorBrewer)
load("cbWomen.rda")
age = sapply(womenMat,
             function(x) suppressWarnings(as.numeric(x[ , 'ag'])))

boxplot(age, main = "Age by Year for Female Runners",
        ylab = "Age", xlab = "Year")
```



Scatter Plot for Run Times vs. Age for Female Runners

Solution:

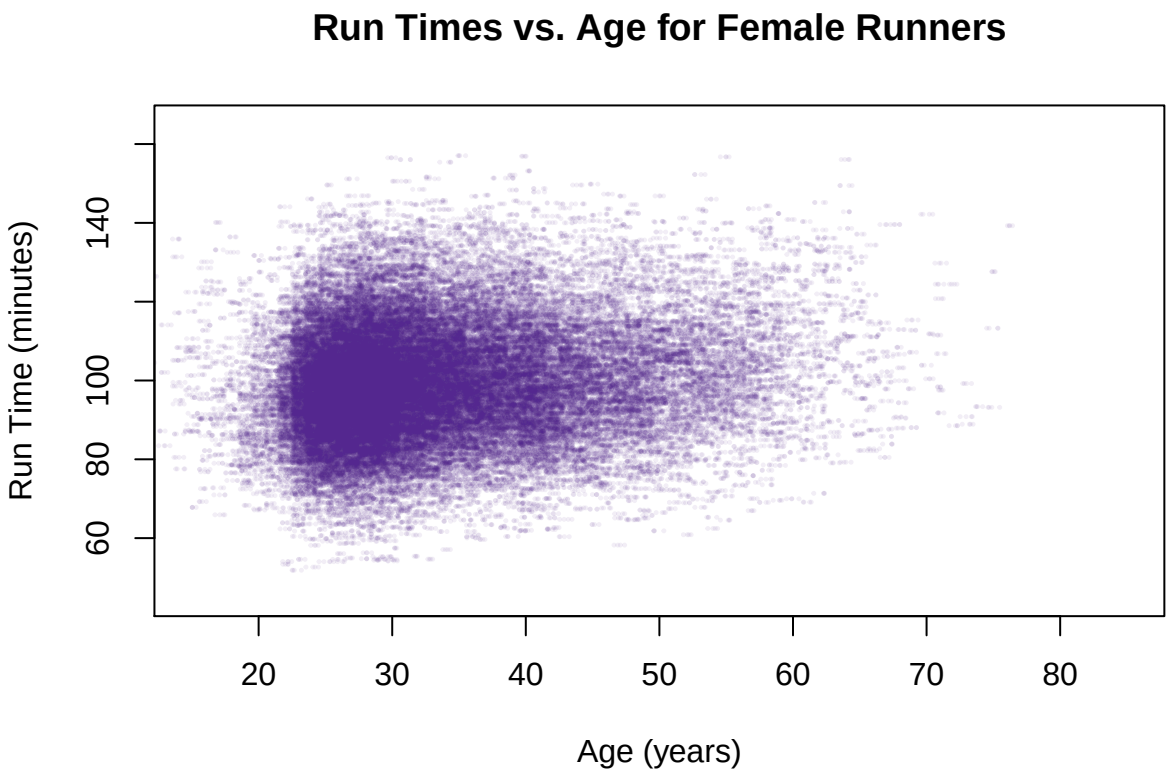
```
load("cbWomen.rda")

plot(runTime ~ age, data = cbWomen, ylim = c(40, 180),
     xlab = "Age (years)", ylab = "Run Time (minutes)",
     main = "Run Times vs. Age for Female Runners")
```



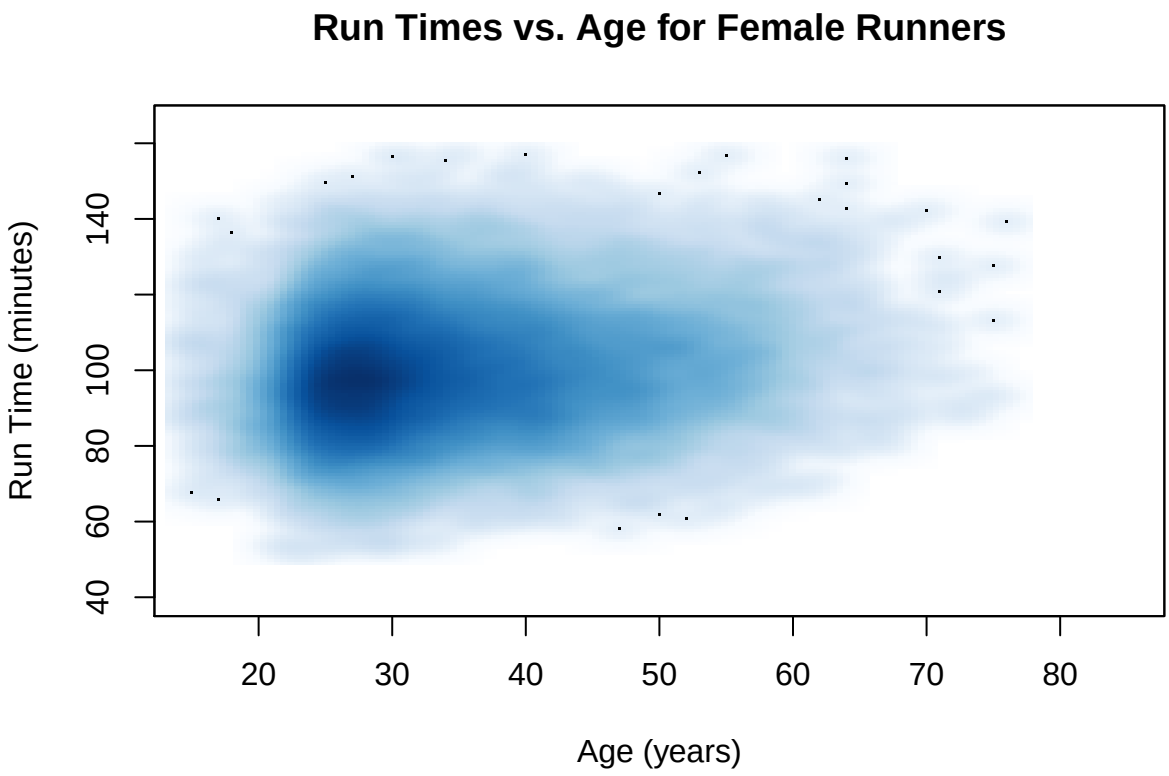
Fit Models to Average Performance

Solution:

```
Purples8 = brewer.pal(9, "Purples")[8]
Purples8A = paste(Purples8, "14", sep = "")

plot(runTime ~ jitter(age, amount = 0.5),
     data = cbWomen,
     pch = 19,cex = 0.2, col = Purples8A,
     ylim = c(45, 165), xlim = c(15, 85),
     xlab = "Age (years)", ylab = "Run Time (minutes)",
     main = "Run Times vs. Age for Female Runners")
```




```
smoothScatter(y = cbWomen$runTime, x = cbWomen$age,
              ylim = c(40, 165), xlim = c(15, 85),
              xlab = "Age (years)", ylab = "Run Time (minutes)",
              main = "Run Times vs. Age for Female Runners")
```

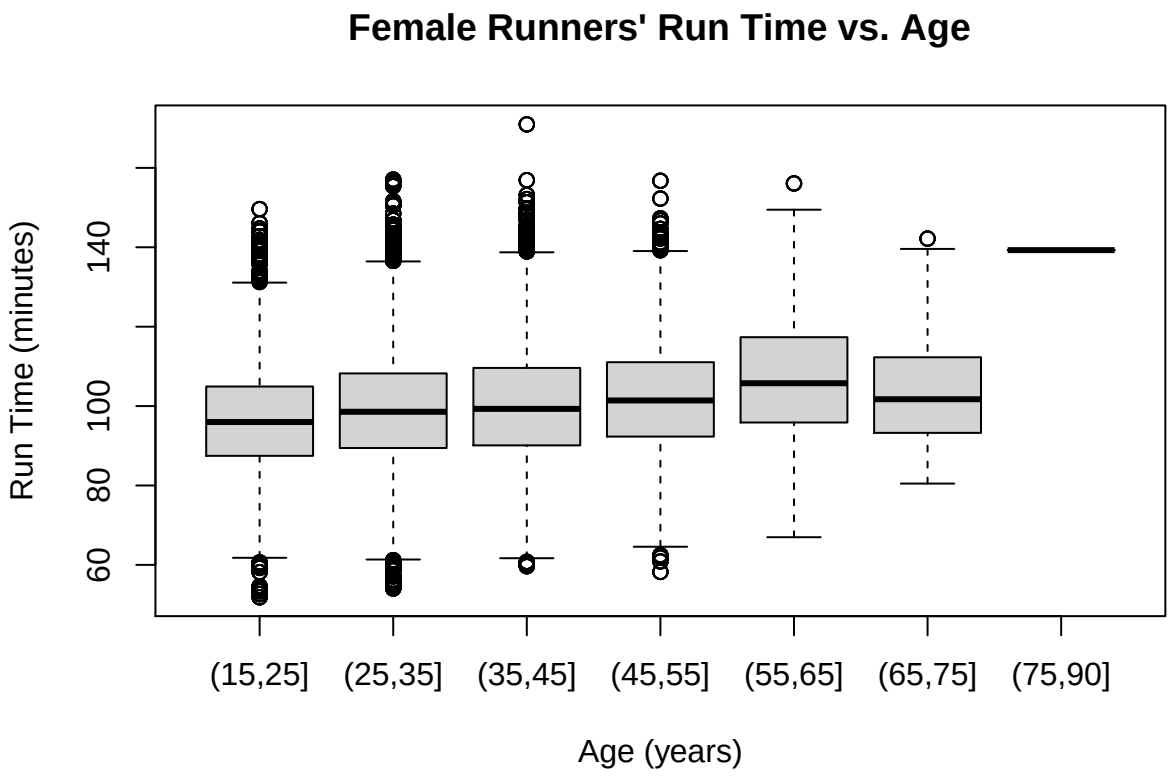


Side-by-Side Boxplots of Female Runners Run Time vs. Age

```
cbWomenSub = cbWomen [cbWomen$runTime>30 & cbWomen$age>15,]
ageCat = cut(cbWomenSub$age, breaks=c(seq(15, 75, 10), 90))
table(ageCat)
```

```
## ageCat
## (15,25] (25,35] (35,45] (45,55] (55,65] (65,75] (75,90]
##   18660   54064   24124   10172    2952     288       4
```

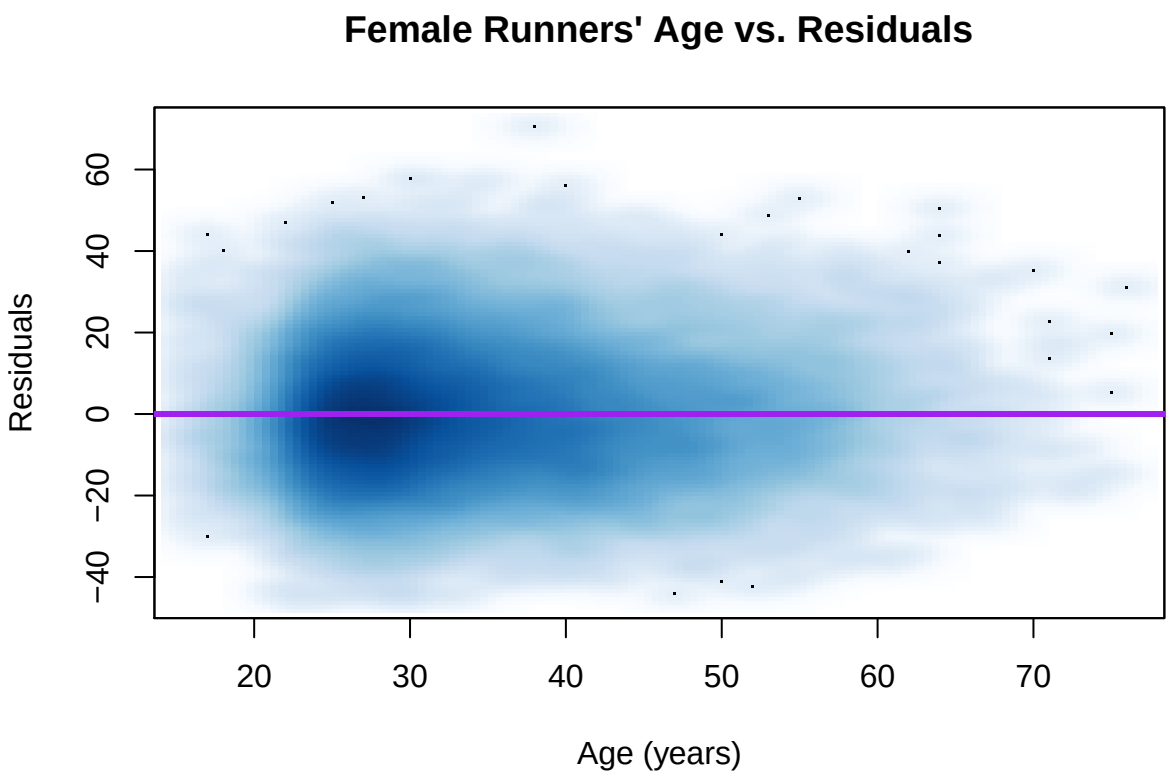
```
plot(cbWomenSub$runTime~ageCat, main="Female Runners' Run Time vs. Age",
     xlab="Age (years)", ylab="Run Time (minutes)")
```



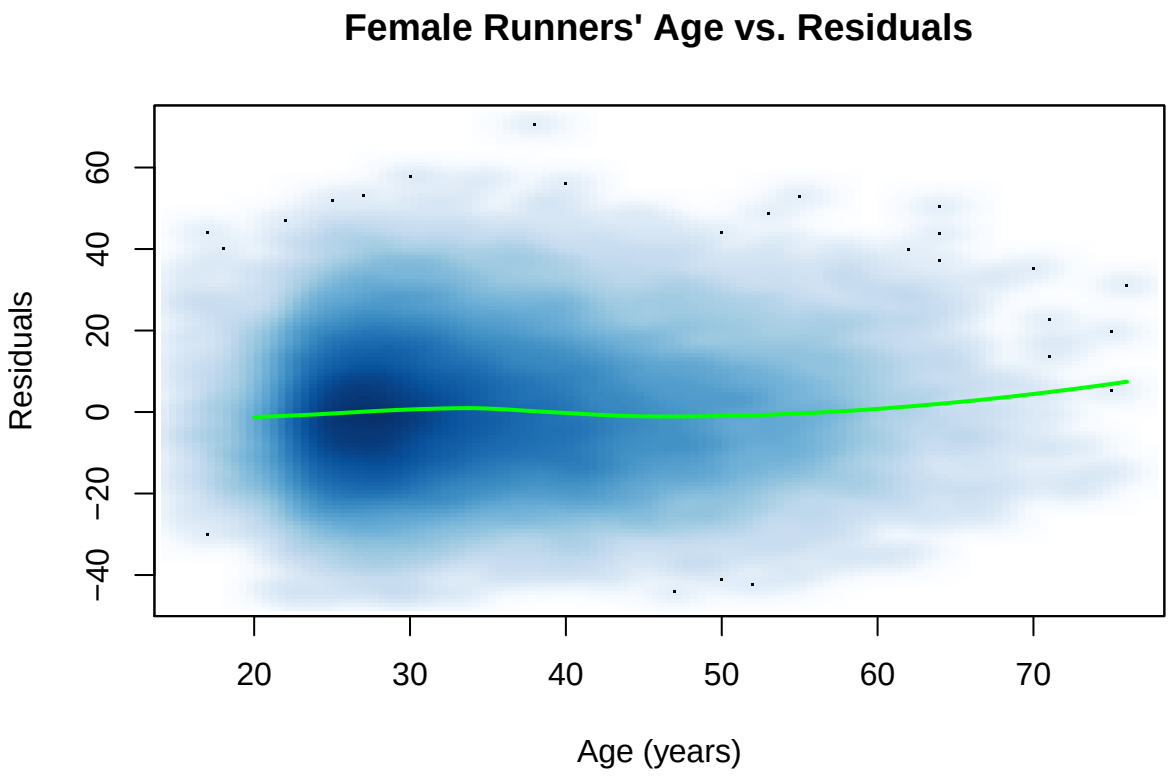
Residual Plot from Fitting a Simple Linear Model of Performance to Age

Solution:

```
lmAge = lm(runTime ~ age, data = cbWomenSub)

cbWomenSubAge = cbWomenSub$age[1:length(lmAge$residuals)]
smoothScatter(x = cbWomenSubAge, y = lmAge$residuals,
              xlab = "Age (years)", ylab = "Residuals",
              main = "Female Runners' Age vs. Residuals")
abline(h = 0, col = "purple", lwd = 3)
```




```
resid.lo = loess(resids ~ age, data = data.frame(resids = residuals(lmAge),
                                                 age = cbWomenSubAge))
age20to80 = 20:80

resid.lo.pr = predict(resid.lo, newdata = data.frame(age = age20to80))

smoothScatter(x = cbWomenSubAge, y = lmAge$residuals,
              xlab = "Age (years)", ylab = "Residuals",
              main = "Female Runners' Age vs. Residuals")+
  lines(x = age20to80, y = resid.lo.pr, col = "green", lwd = 2)
```



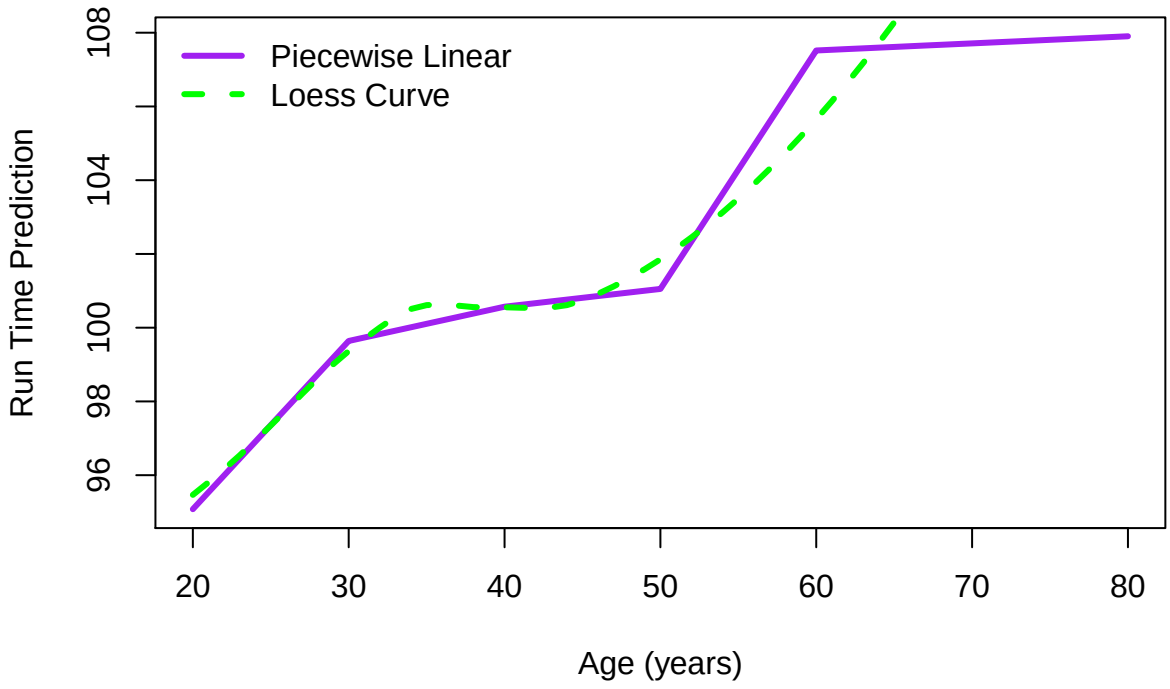

```
## integer(0)
```

Piecewise Linear and Loess Curves Fitted to Run Time vs. Age

Solution:

```
womenRes.lo = loess(runTime ~ age, cbWomenSub)
womenRes.lo.pr = predict(womenRes.lo, data.frame(age = age20to80))
over50 = pmax(0, cbWomenSub$age - 50)
lmOver50 = lm(runTime ~ age + over50, data = cbWomenSub)
# summary(lmOver50)

decades = seq(30, 60, by = 10)
overAge = lapply(decades, function(x) pmax(0, (cbWomenSub$age - x)))
names(overAge) = paste("over", decades, sep = "")
overAge = as.data.frame(overAge)

lmPiecewise = lm(runTime ~ . , data = cbind(cbWomenSub[, c("runTime", "age")], overAge))
# summary(lmPiecewise)

overAge20 = lapply(decades, function(x) pmax(0, (age20to80 - x)))
names(overAge20) = paste("over", decades, sep = "")
overAgeDF = cbind(age = data.frame(age = age20to80), overAge20)

predPiecewise = predict(lmPiecewise, overAgeDF)
```

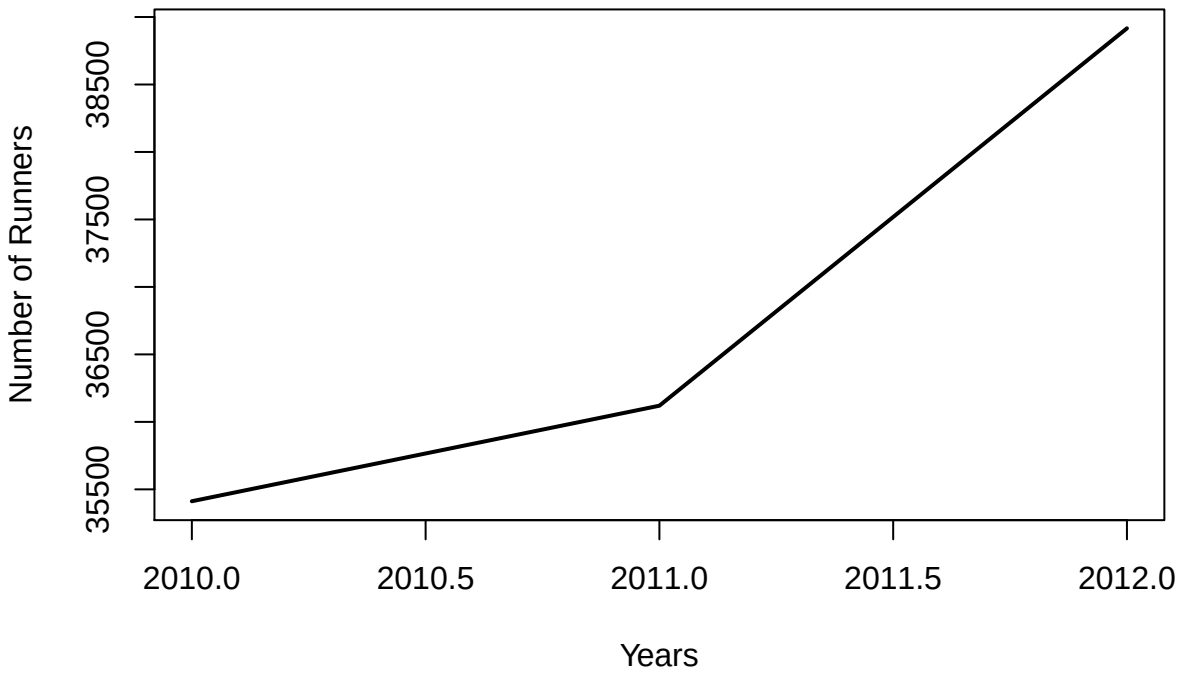
```
plot(predPiecewise ~ age20to80,
     type = "l", col = "purple", lwd = 3,
     xlab = "Age (years)", ylab = "Run Time Prediction")
lines(x = age20to80, y = womenRes.lo.pr,
      col = "green", lty = 2, lwd = 3)
legend("topleft", col = c("purple", "green"),
       lty = c(1, 2), lwd= 3,
       legend = c("Piecewise Linear", "Loess Curve"), bty = "n")
```



Line Plot of the Number of Female Runners by Year

Solution:

```
numRunners = with(cbWomen, tapply(runTime, year, length))
plot(numRunners ~ names(numRunners), type="l", lwd = 2,
     xlim = c(2010,2012),
     xlab = "Years", ylab = "Number of Runners")
```



Density Curves for the Age of Female Runners for 2 years (smallest and largest year that you analyzed)

Solution:

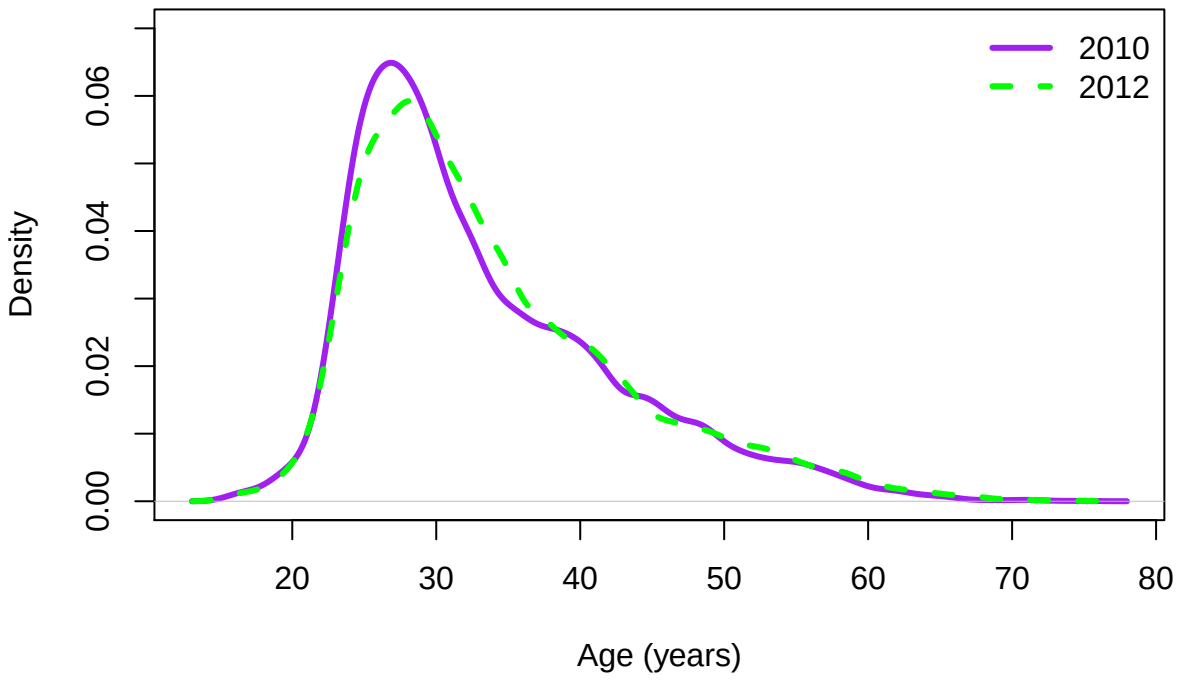
```
summary(cbWomenSub$runTime[cbWomenSub$year == 2010])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   51.85   89.77   98.85   99.71  108.82  157.05
```

```
summary(cbWomenSub$runTime[cbWomenSub$year == 2012])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   54.03   89.08   98.03   99.02  107.90  170.97
```

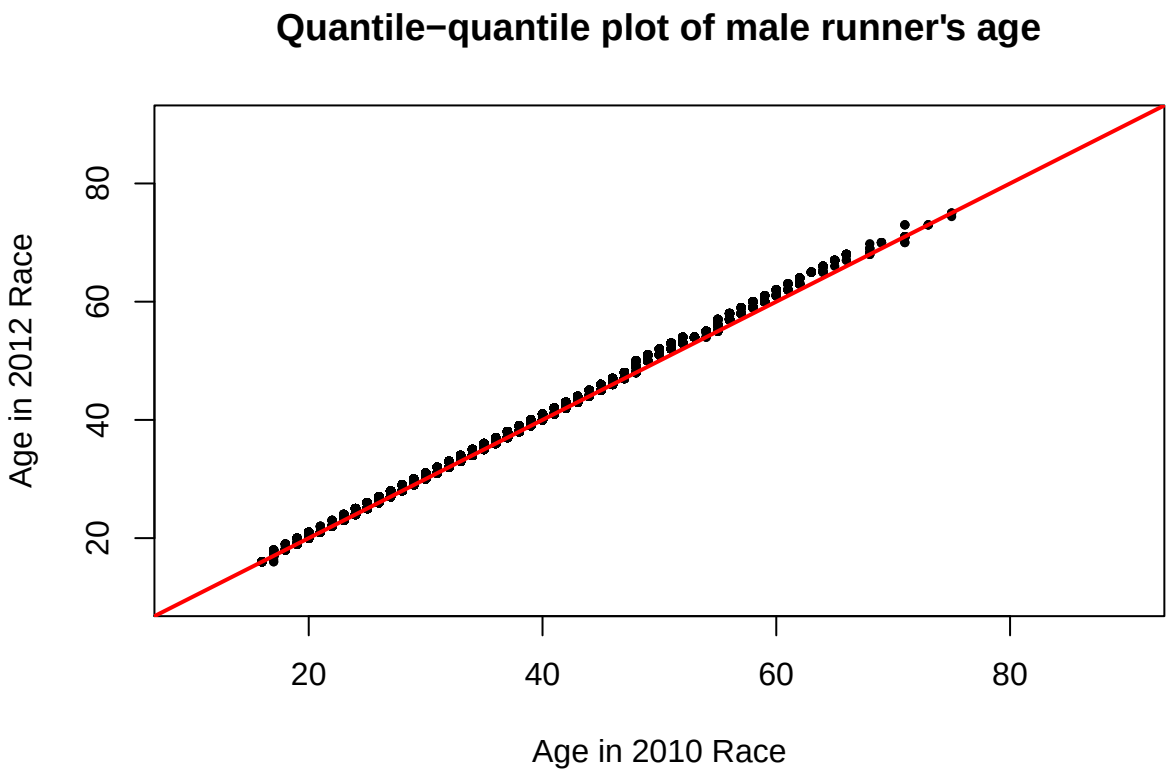
```
age2010 = cbWomenSub[ cbWomenSub$year == 2010, "age" ]
age2012 = cbWomenSub[ cbWomenSub$year == 2012, "age" ]
plot(density(age2010, na.rm = TRUE),
     ylim = c(0, 0.07), col = "purple",
     lwd = 3, xlab = "Age (years)", main = "")
lines(density(age2012, na.rm = TRUE),
      lwd = 3, lty = 2, col="green")
legend("topright", col = c("purple", "green"), lty= 1:2, lwd = 3,
       legend = c("2010", "2012"), bty = "n")
```



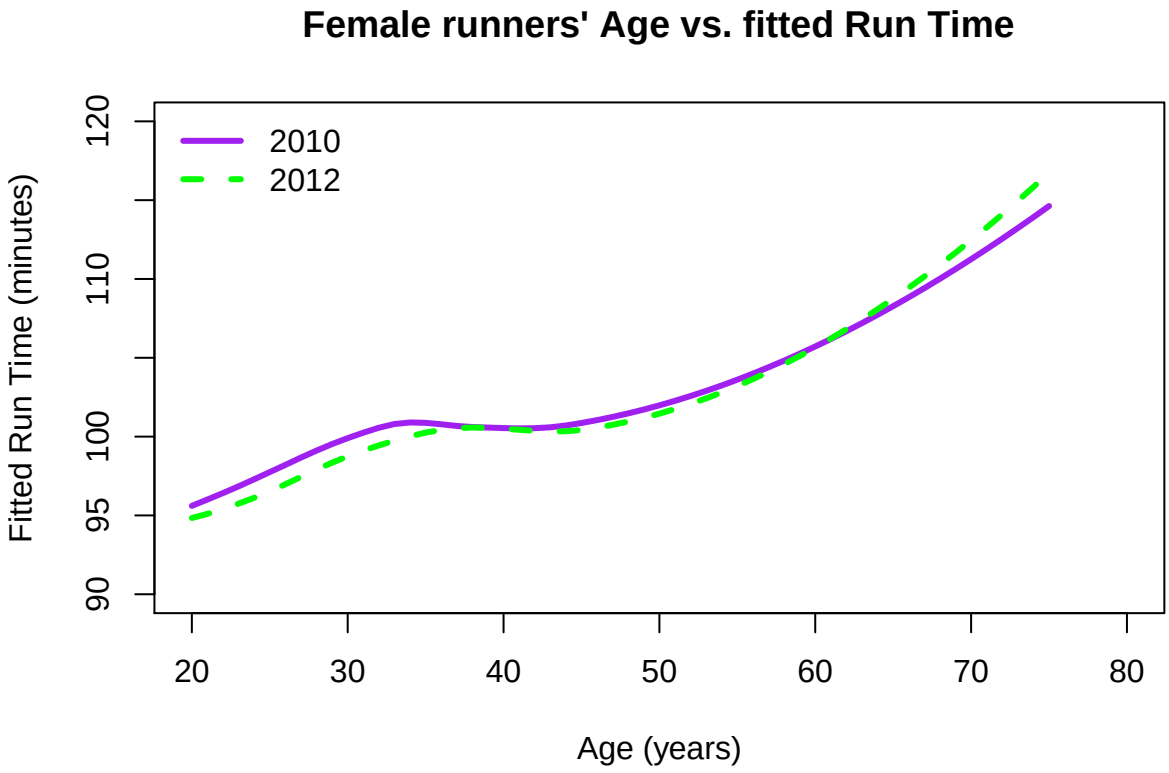
Loess Curves Fit to Performance for 2 years (smallest and largest year that you analyzed) Female Runners

Solution:

```
qqplot(age2010, age2012, pch = 19, cex = 0.5,
       ylim = c(10,90), xlim = c(10,90),
       xlab = "Age in 2010 Race",
       ylab = "Age in 2012 Race",
       main = "Quantile-quantile plot of male runner's age")
abline(a = 0, b = 1, col="red", lwd = 2)
```



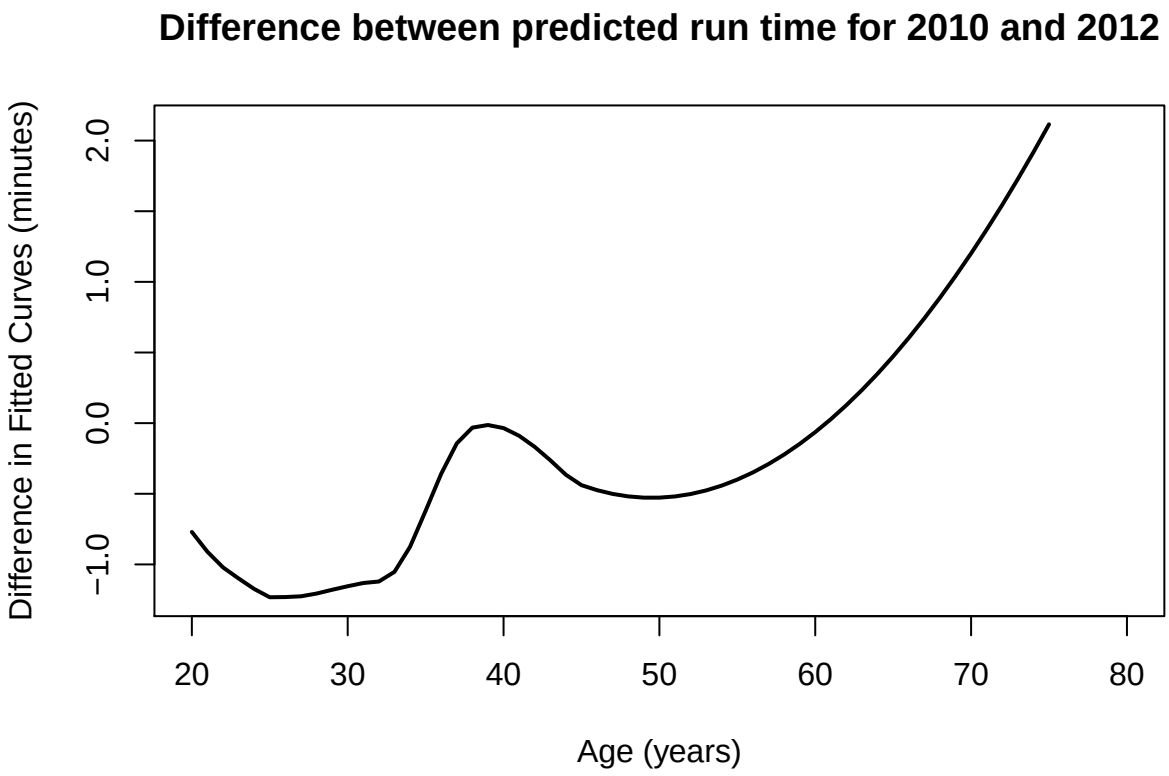

```
mR.lo01 = loess(runTime ~ age, cbWomenSub[ cbWomenSub$year == 2010,])
mR.lo.pr01 = predict(mR.lo01, data.frame(age = age20to80))
mR.lo12 = loess(runTime ~ age, cbWomenSub[ cbWomenSub$year == 2012,])
mR.lo.pr12 = predict(mR.lo12, data.frame(age = age20to80))
plot(mR.lo.pr01 ~ age20to80, ylim = c(90,120), xlim = c(20,80),
     type = "l", col = "purple", lwd = 3,
     xlab = "Age (years)", ylab = "Fitted Run Time (minutes)",
     main = "Female runners' Age vs. fitted Run Time")
lines(x = age20to80, y = mR.lo.pr12,
      col = "green", lty = 2, lwd = 3)
legend("topleft", col = c("purple", "green"), lty = 1:2, lwd = 3,
       legend = c("2010", "2012"), bty = "n")
```



Difference between Loess Curves of the predicted run time for 2 years (smallest and largest year that you analyzed)

Solution:

```
gap11 = mR.lo.pr12 - mR.lo.pr01
plot(gap11 ~ age20to80, type = "l" , xlab = "Age (years)",
     ylab = "Difference in Fitted Curves (minutes)",
     main = "Difference between predicted run time for 2010 and 2012", lwd = 2)
```



Perform comparative analysis of the performance of the male runners (previously analyzed in class) and female runners for the yearly data that you selected

Solution:

For Men runners,the Run time is concentrated between 40 and 160 min whereas for Female runners, the Run time is concentrated between 20 and 160 min.

Also,the number of Men runners for the years 2010 to 2012 is higher (above 6000) and is greater than Female runners for years 2010 to 2012 as Number of Female runners is below 5000.

The median Run Time is around 100 min for Female runners for years 2010 to 2012 whereas the median Run time is around 85 for Male runners for the same years