HomeWork-2

Viveksinh

2/24/2020

Female Runners' analysis from years 2010 to 2012

```
urls = paste(ubase, "results/", 2010:2012, "/",
             2010:2012, "cucb10m-f.htm", sep = "")

extractResTable =
  # Retrieve data from web site,
  # find preformatted text,
  # return as a character vector.
  function(url)
  {
    doc = htmlParse(url)
    preNode = getNodeSet(doc, "//pre")
    txt = xmlValue(preNode[[1]])
    els = strsplit(txt, "\r\n")[[1]]

    return(els)
  }

womenTables = lapply(urls, extractResTable)

names(womenTables) = 2010:2012

sapply(womenTables, length)
```

```
## 2010 2011 2012
## 8863 9038 9737
```

```
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))
}

selectCols = function(shortColNames, headerRow, searchLocs) {
  sapply(shortColNames, function(shortName, headerRow, searchLocs){
    startPos = regexpr(shortName, headerRow)[[1]]
    if (startPos == -1) return( c(NA, NA) )
```

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

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

    # Find the index of the row with =s
    eqIndex = grep("^===", file)
    # Extract the two key rows and the data
    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

    return(Values)
  }

womenResMat = lapply(womenTables, extractVariables)
length(womenResMat)
```

```
## [1] 3
```

```
# returns the number of rows present
sapply(womenResMat, nrow)
```

```
## 2010 2011 2012
## 8853 9030 9729
```

```
age = sapply(womenResMat,function(x) as.numeric(x[ , 'ag']))

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

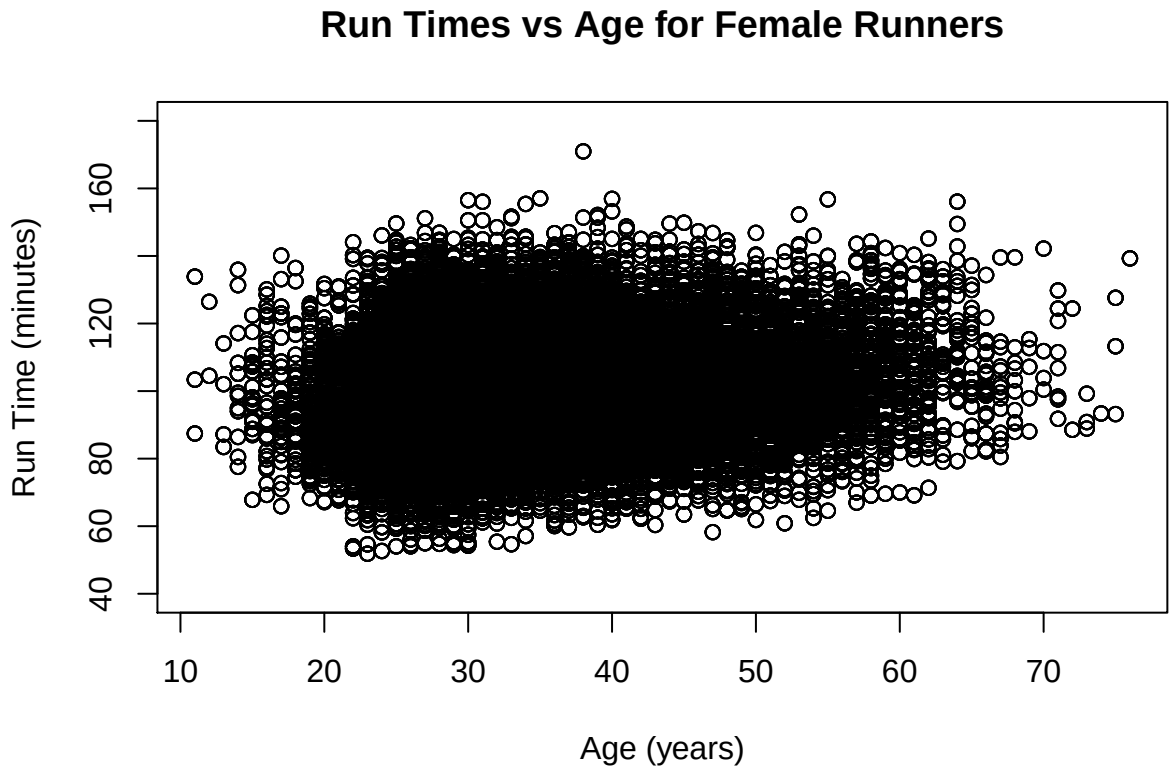

```
convertTime = function(time) {
  timePieces = strsplit(time, ":")
  timePieces = sapply(timePieces, as.numeric)
  sapply(timePieces, function(x) {
    if (length(x) == 2) x[1] + x[2]/60
    else 60*x[1] + x[2] + x[3]/60
  })
}

createDF = function(Res, year, sex)
{
  # 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
  Res = Res[ useTime != "", ]

  Results = data.frame(year = rep(year, nrow(Res)),
                       sex = rep(sex, nrow(Res)),
                       name = Res[ , 'name'], home = Res[ , 'home'],
```

```
                        age = as.numeric(Res[, 'ag']),
                        runTime = runTime,
                        stringsAsFactors = FALSE)
  invisible(Results)
}

womenDF = mapply(createDF, womenResMat, year = 2010:2012,
                 sex = rep("F", 14), SIMPLIFY = FALSE)
```

```
## Warning in mapply(createDF, womenResMat, year = 2010:2012, sex = rep("F", :
## longer argument not a multiple of length of shorter
```

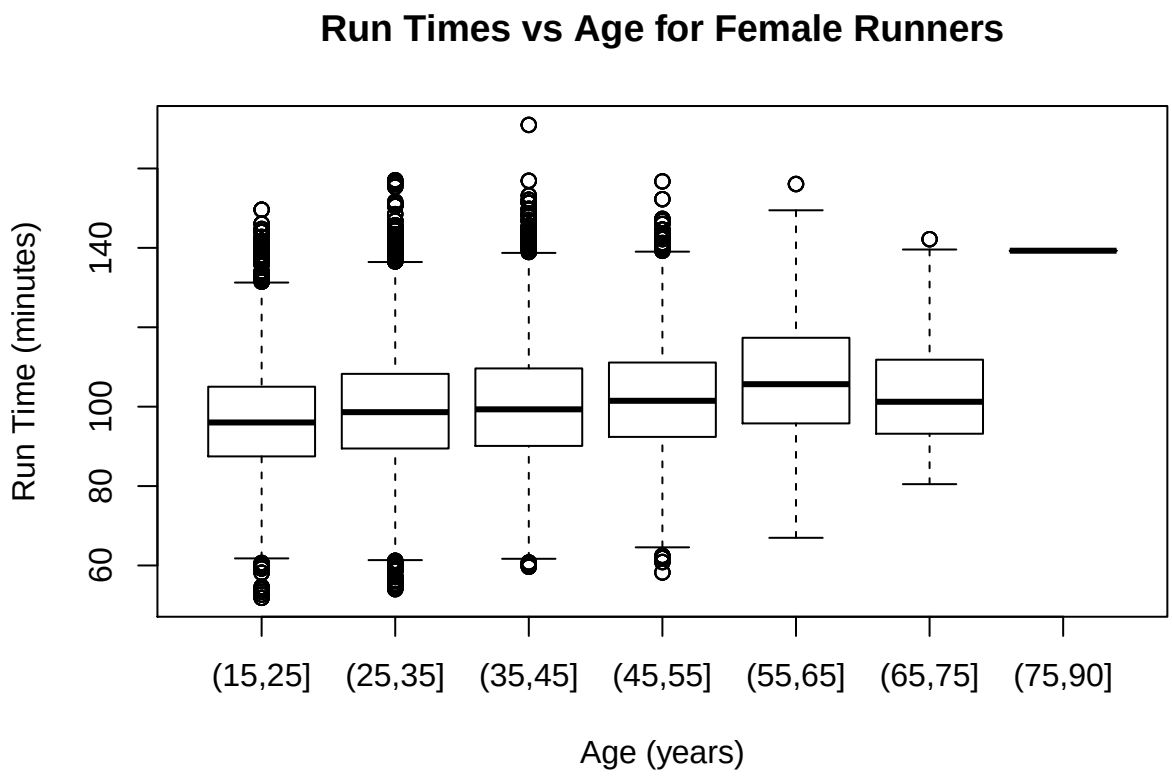
```
## Warning in mapply(createDF, womenResMat, year = 2010:2012, sex = rep("F", :
## longer argument not a multiple of length of shorter
```

```
cbWomen = do.call(rbind, womenDF)

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

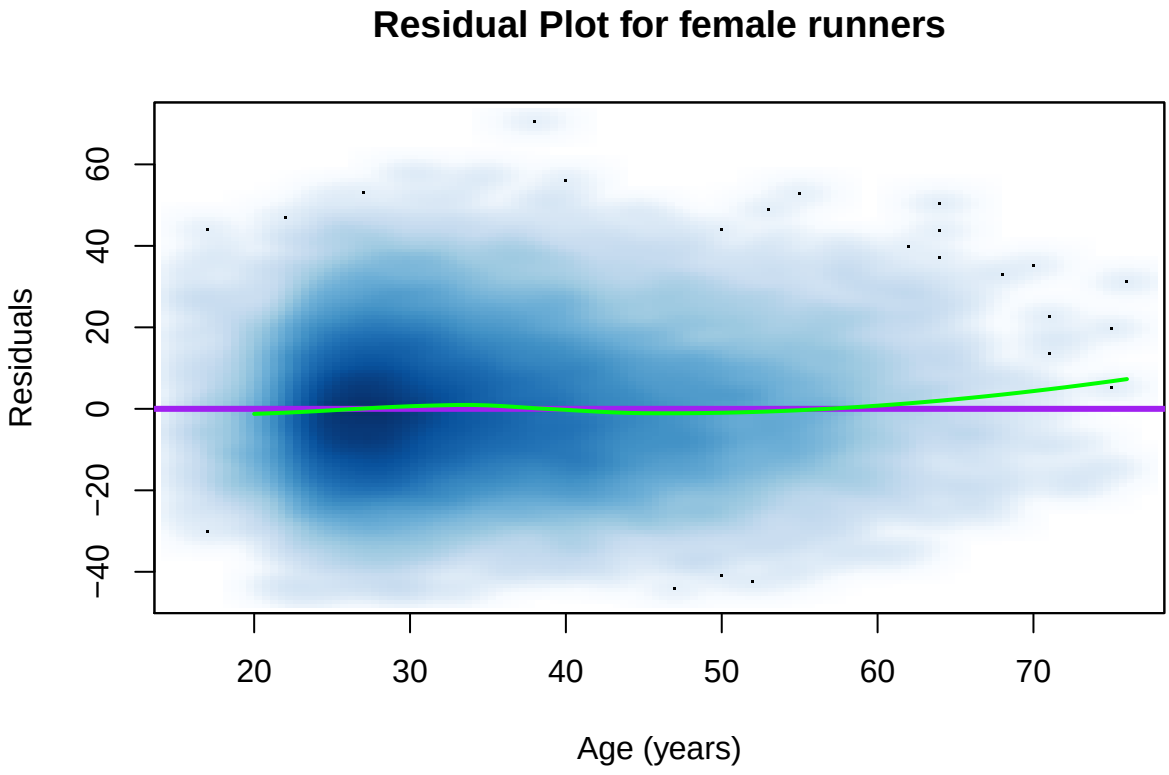



```
## ********************************************************************* ##
cbWomenSub = cbWomen[cbWomen$runTime > 30 &
          !is.na(cbWomen$age) & cbWomen$age > 15, ]

ageCat = cut(cbWomenSub$age, breaks = c(seq(15, 75, 10), 90))
```

```r
## Side-by-Side Boxplots of Female Runners' Run Time vs. Age
plot(cbWomenSub$runTime ~ ageCat,
     xlab = "Age (years)", ylab = "Run Time (minutes)",
     main="Run Times vs Age for Female Runners")
```

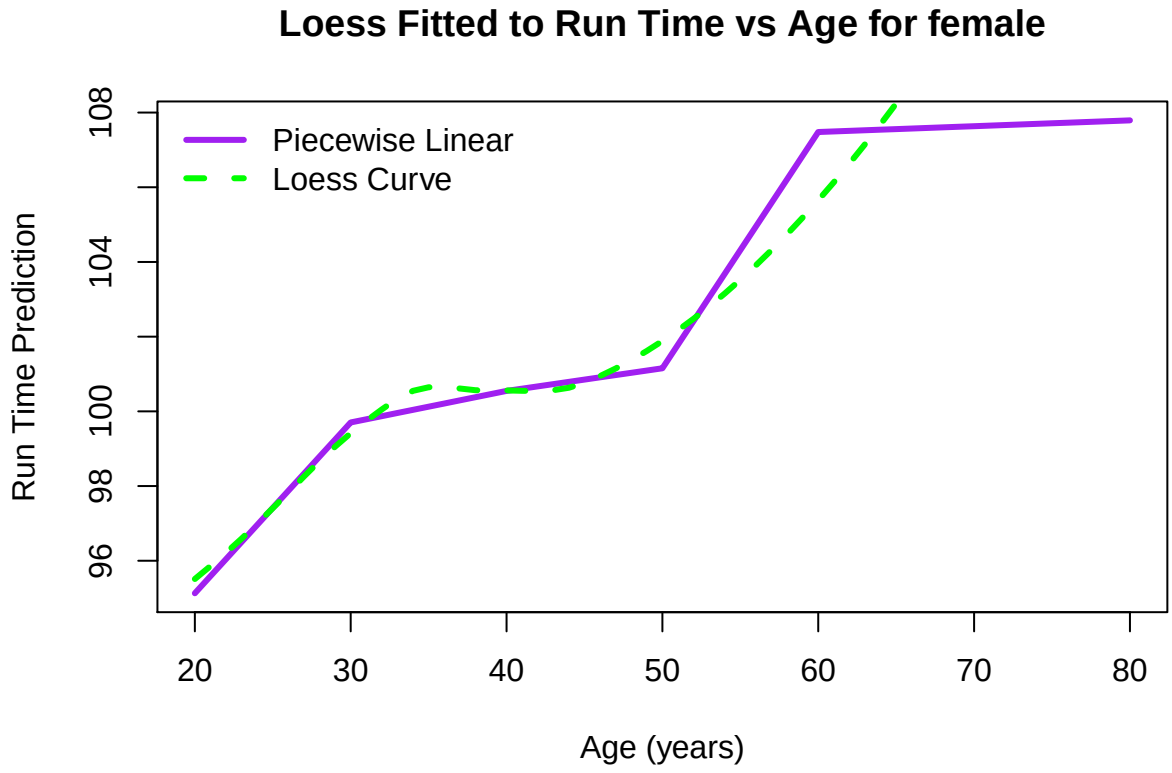



```r
lmAge = lm(runTime ~ age, data = cbWomenSub)

## Residual Plot from Fitting a Simple Linear Model of Performance to Age
smoothScatter(x = cbWomenSub$age, y = lmAge$residuals,
              xlab = "Age (years)", ylab = "Residuals",
              main="Residual Plot for female runners")
abline(h = 0, col = "purple", lwd = 3)

# local polynomial regression
resid.lo = loess(resids ~ age, data = data.frame(resids = residuals(lmAge),age = cbWomenSub$age))

age20to80 = 20:80

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

lines(x = age20to80, y = resid.lo.pr, col = "green", lwd = 2)
```

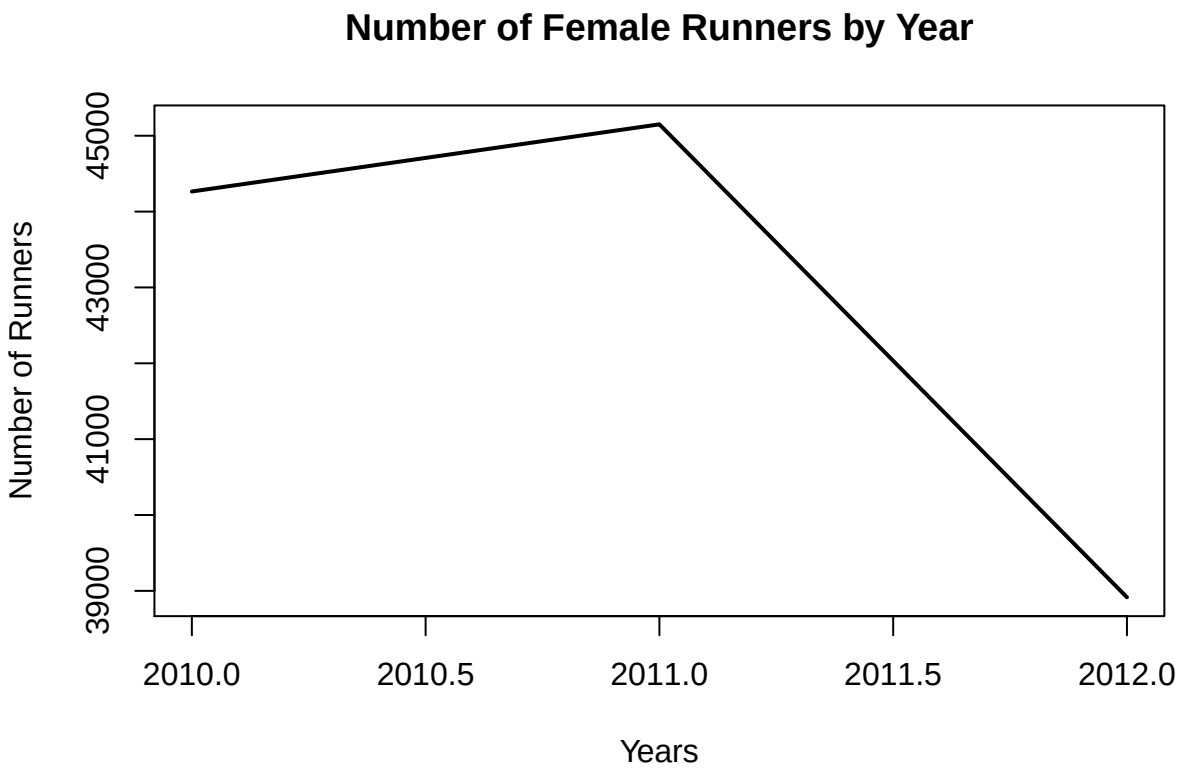



```
menRes.lo = loess(runTime ~ age, cbWomenSub)

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

over50 = pmax(0, cbWomenSub$age - 50)

lmOver50 = lm(runTime ~ age + over50, data = cbWomenSub)

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))

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)

## Loess Curves Fitted to Run Time vs. Age
plot(predPiecewise ~ age20to80,
```

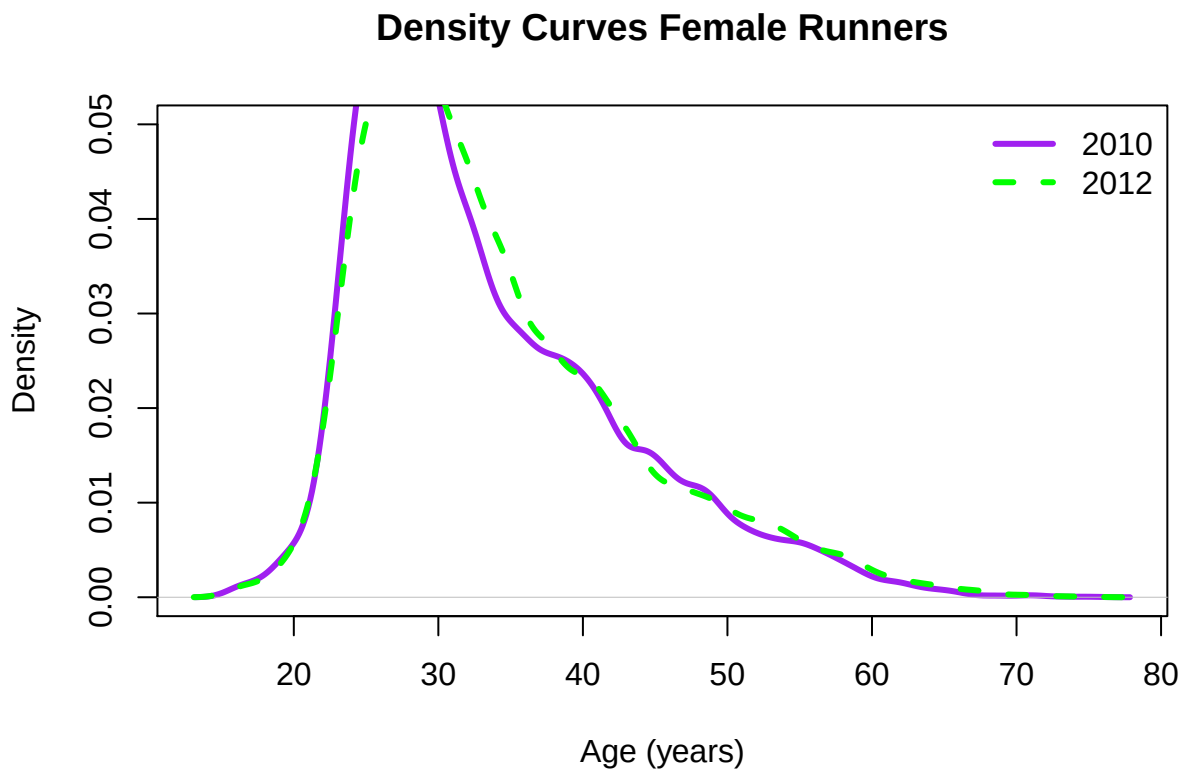
```
     type = "l", col = "purple", lwd = 3,
     xlab = "Age (years)", ylab = "Run Time Prediction",
     main= "Loess Fitted to Run Time vs Age for female")

lines(x = age20to80, y = menRes.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")
```



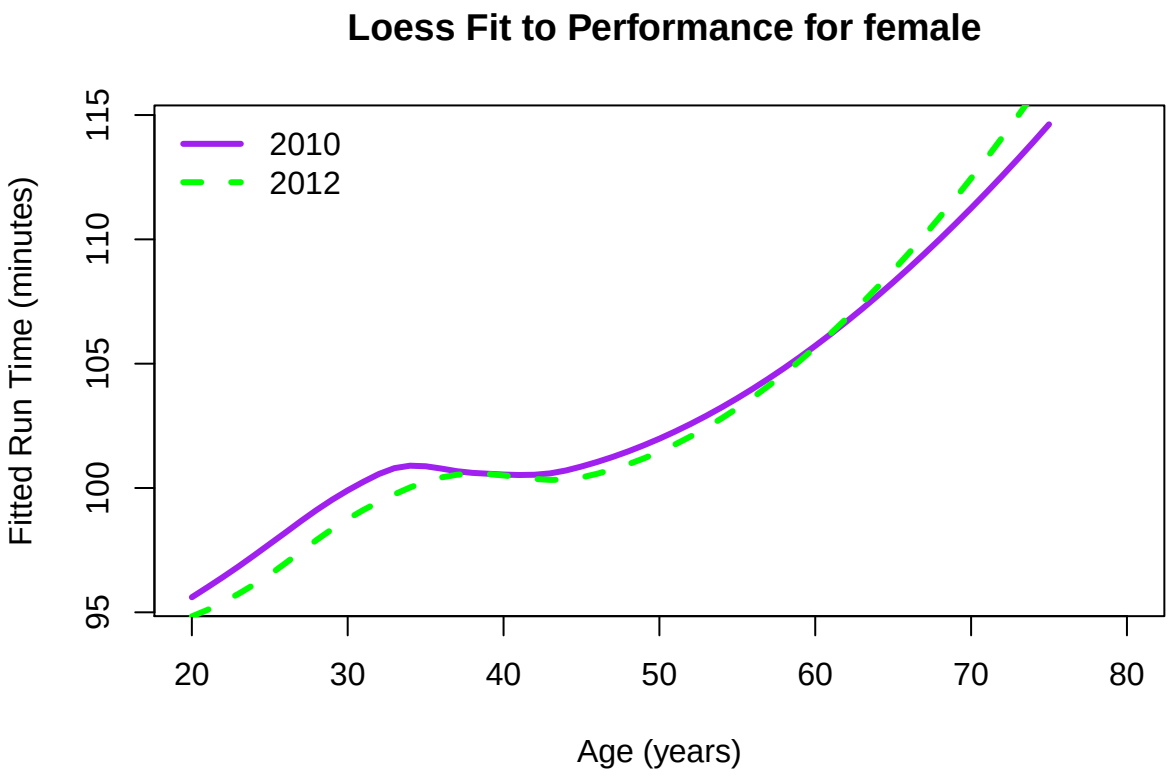

```
## Line Plot of the Number of Female Runners by Year
numRunners = with(cbWomen, tapply(runTime, year, length))
plot(numRunners ~ names(numRunners), type="l", lwd = 2,
     xlab = "Years", ylab = "Number of Runners",
     main="Number of Female Runners by Year")
```



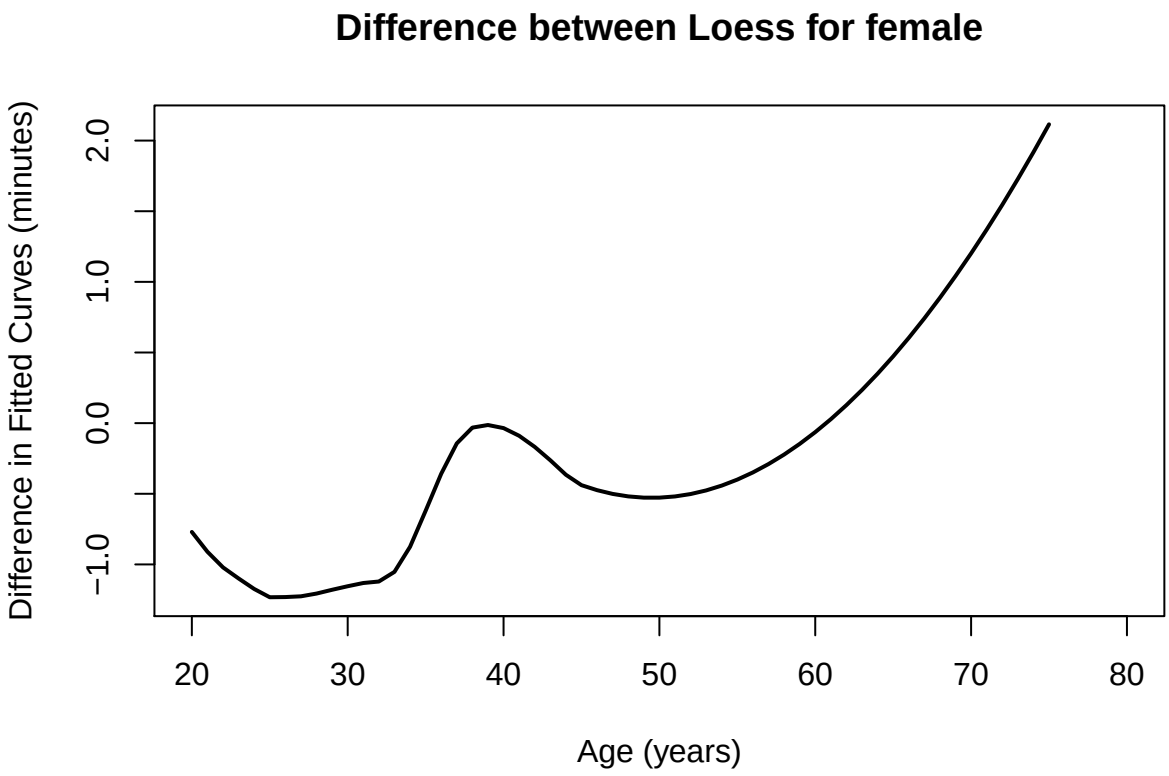

```
age2010 = cbWomenSub[ cbWomenSub$year == 2010, "age" ]
age2012 = cbWomenSub[ cbWomenSub$year == 2012, "age" ]

## Density Curves for the Age of Female Runners for 2 years
plot(density(age2010, na.rm = TRUE),
     ylim = c(0, 0.05), col = "purple",
     lwd = 3,  xlab = "Age (years)",
     main = "Density Curves Female Runners")

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")
```




```
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))

## Loess Curves Fit to Performance for 2 years
plot(mR.lo.pr01 ~ age20to80,
     type = "l", col = "purple", lwd = 3,
     xlab = "Age (years)", ylab = "Fitted Run Time (minutes)",
     main=" Loess Fit to Performance for female")

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")
```




```
gap12 = mR.lo.pr12 - mR.lo.pr01

## Difference between Loess Curves of the predicted run time for 2 years
plot(gap12 ~ age20to80, type = "l" , xlab = "Age (years)",
     ylab = "Difference in Fitted Curves (minutes)", lwd = 2,
     main="Difference between Loess for female")
```



Male Runners' analysis from years 2010 to 2012

```
##################################
urls = paste(ubase, "results/", 2010:2012, "/",
             2010:2012, "cucb10m-m.htm", sep = "")

extractResTable =
  # Retrieve data from web site,
  # find preformatted text,
  # return as a character vector.
  function(url)
  {
    doc = htmlParse(url)
    preNode = getNodeSet(doc, "//pre")
    txt = xmlValue(preNode[[1]])
    els = strsplit(txt, "\r\n")[[1]]

    return(els)
  }

menTables = lapply(urls, extractResTable)

names(menTables) = 2010:2012

sapply(menTables, length)
```

```
## 2010 2011 2012
## 6919 7019 7201
```

```
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))
}

selectCols = function(shortColNames, headerRow, searchLocs) {
  sapply(shortColNames, function(shortName, headerRow, searchLocs){
    startPos = regexpr(shortName, headerRow)[[1]]
    if (startPos == -1) return( c(NA, NA) )
    index = sum(startPos >= searchLocs)
    c(searchLocs[index] + 1, searchLocs[index + 1])
  }, headerRow = headerRow, searchLocs = searchLocs )
}

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

    # Find the index of the row with =s
    eqIndex = grep("^===", file)
    # Extract the two key rows and the data
    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

    return(Values)
  }


menResMat = lapply(menTables, extractVariables)
length(menResMat)
```

```
## [1] 3
```

```
# returns the number of rows present
sapply(menResMat, nrow)
```

```
## 2010 2011 2012
## 6909 7011 7193
```

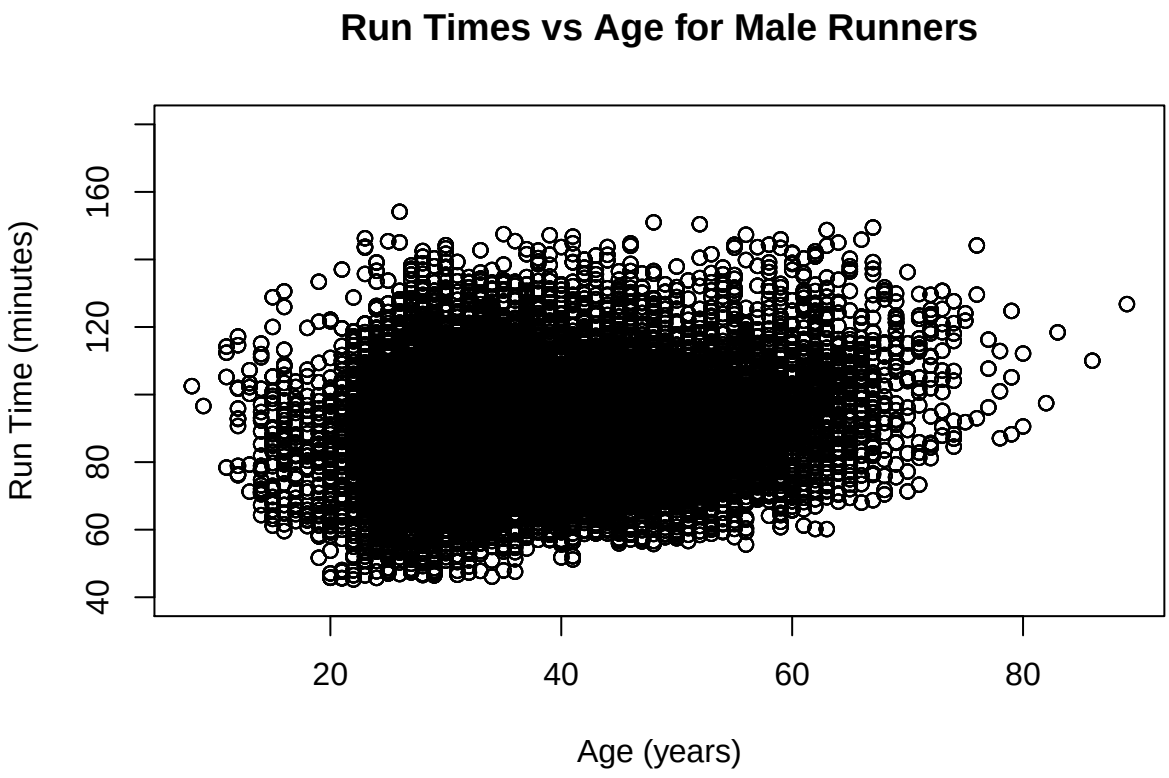
```
age = sapply(menResMat,function(x) as.numeric(x[ , 'ag']))

## Box Plot of Age by Year for Male Runners
boxplot(age, ylab = "Age", xlab = "Year",
        main="Age by Year for Male Runners")
```


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

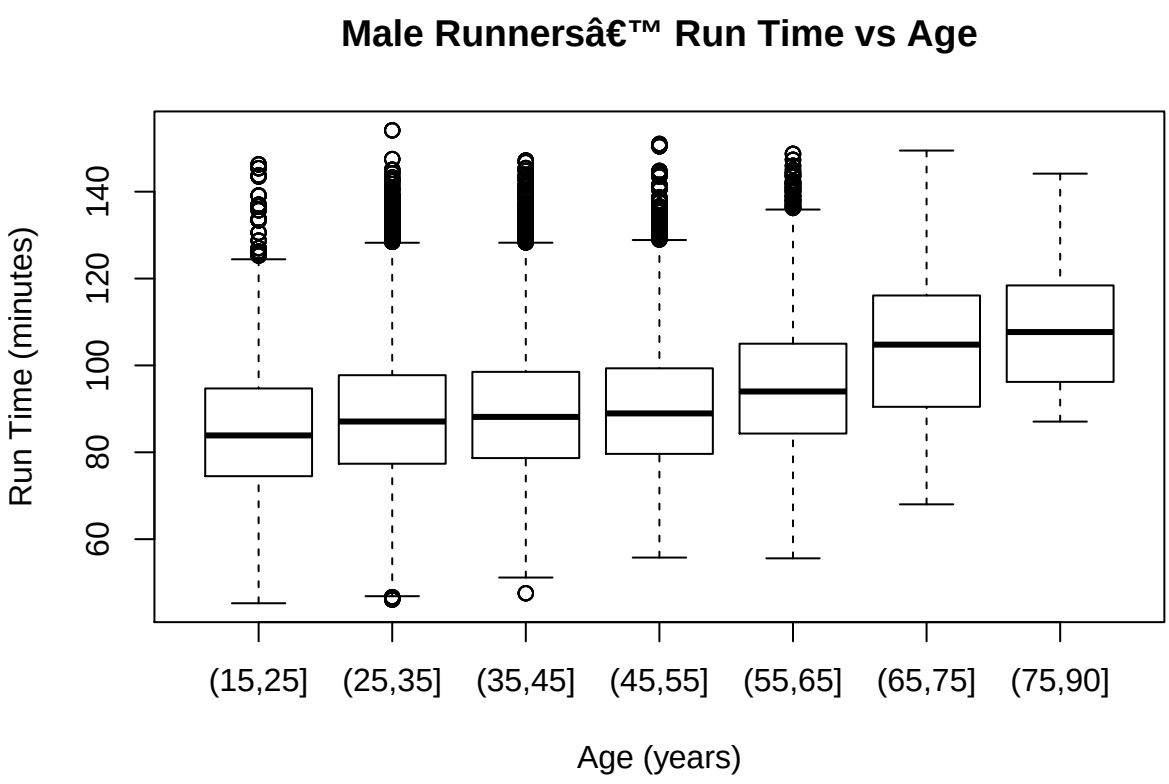
```
  else useTime = Res[ , 'time']

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

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

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

menDF = mapply(createDF, menResMat, year = 2010:2012,
                 sex = rep("F", 14), SIMPLIFY = FALSE)
```

```
## Warning in mapply(createDF, menResMat, year = 2010:2012, sex = rep("F", :
## longer argument not a multiple of length of shorter

## Warning in mapply(createDF, menResMat, year = 2010:2012, sex = rep("F", :
## longer argument not a multiple of length of shorter
```

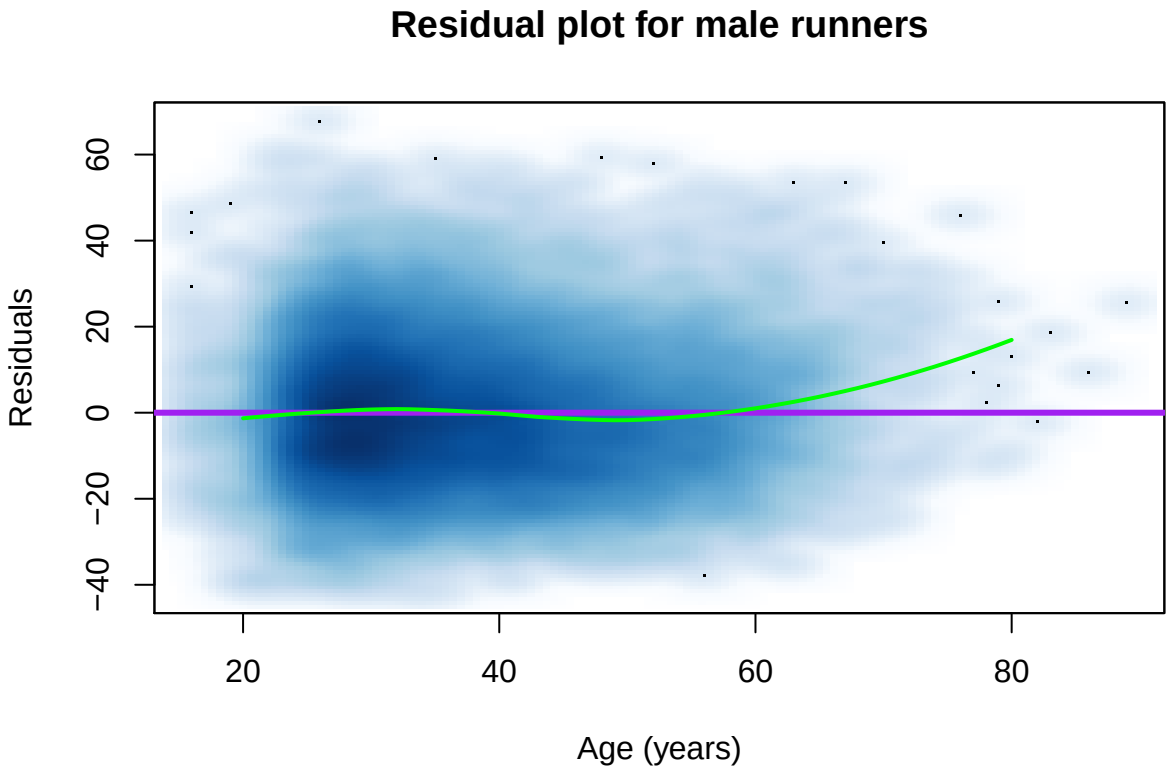
```
cbMen = do.call(rbind, menDF)

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




```
## ********************************************************************** ##
cbMenSub = cbMen[cbMen$runTime > 30 &
                       !is.na(cbMen$age) & cbMen$age > 15, ]

ageCat = cut(cbMenSub$age, breaks = c(seq(15, 75, 10), 90))

## Side-by-Side Boxplots of Male Runners' Run Time vs. Age
plot(cbMenSub$runTime ~ ageCat,
     xlab = "Age (years)", ylab = "Run Time (minutes)",
     main="Male Runners' Run Time vs Age")
```



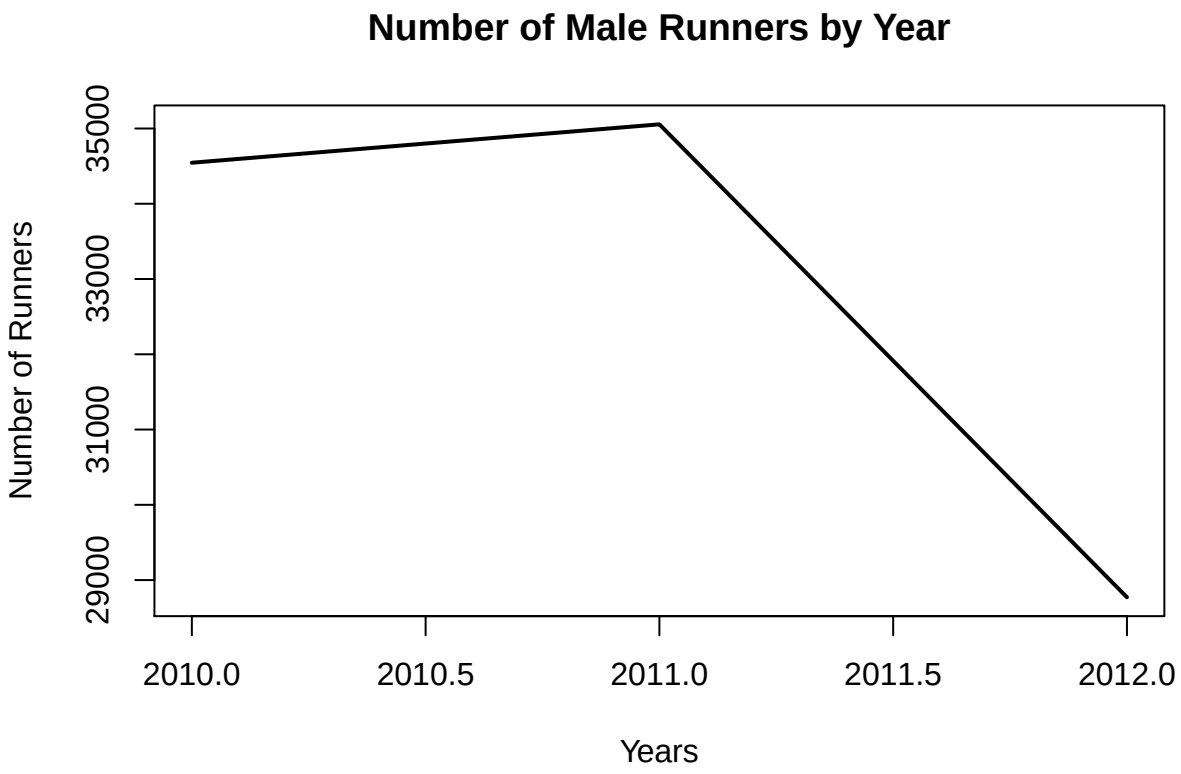

```
lmAge = lm(runTime ~ age, data = cbMenSub)

## Residual Plot from Fitting a Simple Linear Model of Performance to Age
smoothScatter(x = cbMenSub$age, y = lmAge$residuals,
              xlab = "Age (years)", ylab = "Residuals",
              main="Residual plot for male runners")
abline(h = 0, col = "purple", lwd = 3)

# local polynomial regression
resid.lo = loess(resids ~ age, data = data.frame(resids = residuals(lmAge),age = cbMenSub$age))

age20to80 = 20:80

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

lines(x = age20to80, y = resid.lo.pr, col = "green", lwd = 2)
```



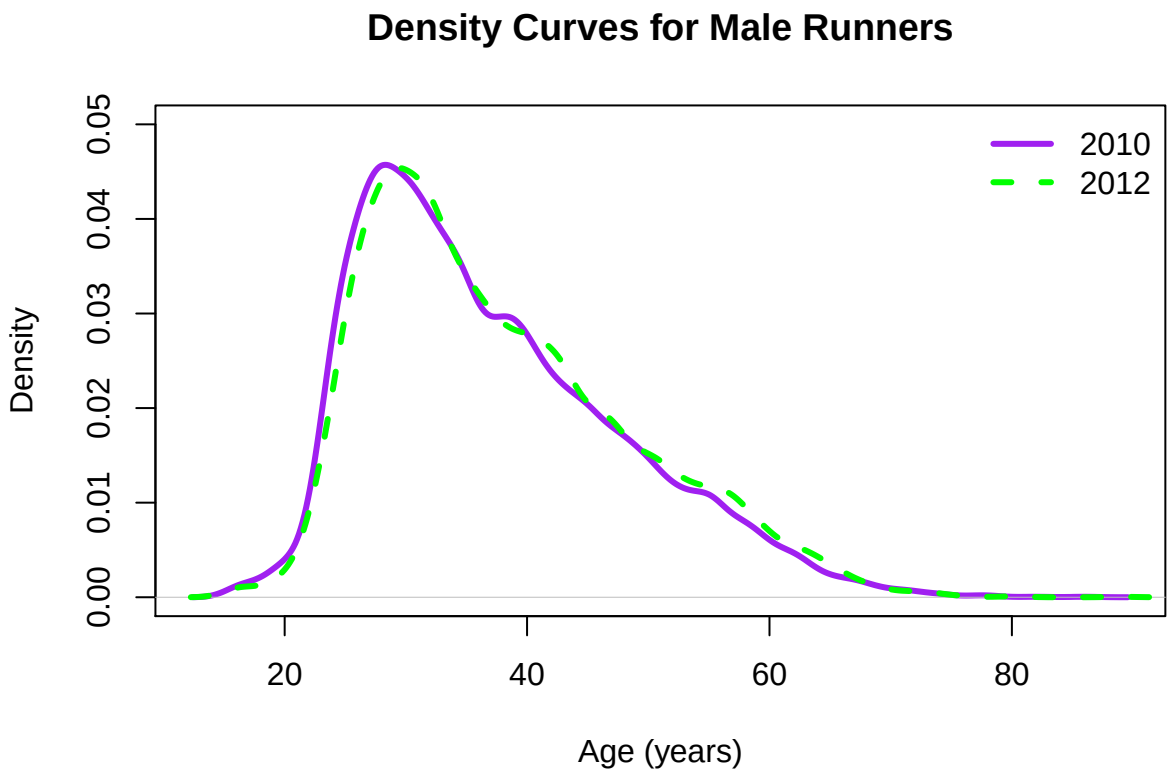

```
menRes.lo = loess(runTime ~ age, cbMenSub)

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

over50 = pmax(0, cbMenSub$age - 50)

lmOver50 = lm(runTime ~ age + over50, data = cbMenSub)

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

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

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)

## Loess Curves Fitted to Run Time vs. Age
plot(predPiecewise ~ age20to80,
```

```
     type = "l", col = "purple", lwd = 3,
     xlab = "Age (years)", ylab = "Run Time Prediction",
     main="Loess Fitted for male runners")

lines(x = age20to80, y = menRes.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")
```

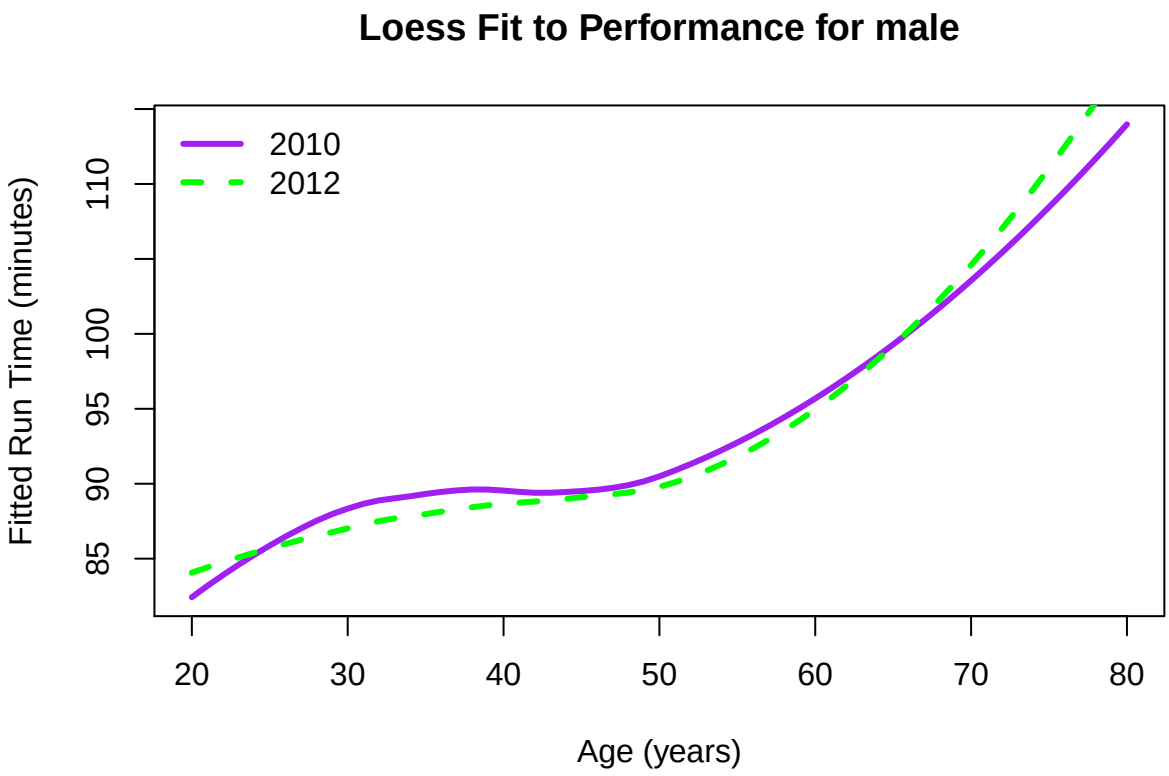

```
## Line Plot of the Number of Male Runners by Year
numRunners = with(cbMen, tapply(runTime, year, length))
plot(numRunners ~ names(numRunners), type="l", lwd = 2,
     xlab = "Years", ylab = "Number of Runners",
     main="Number of Male Runners by Year")
```



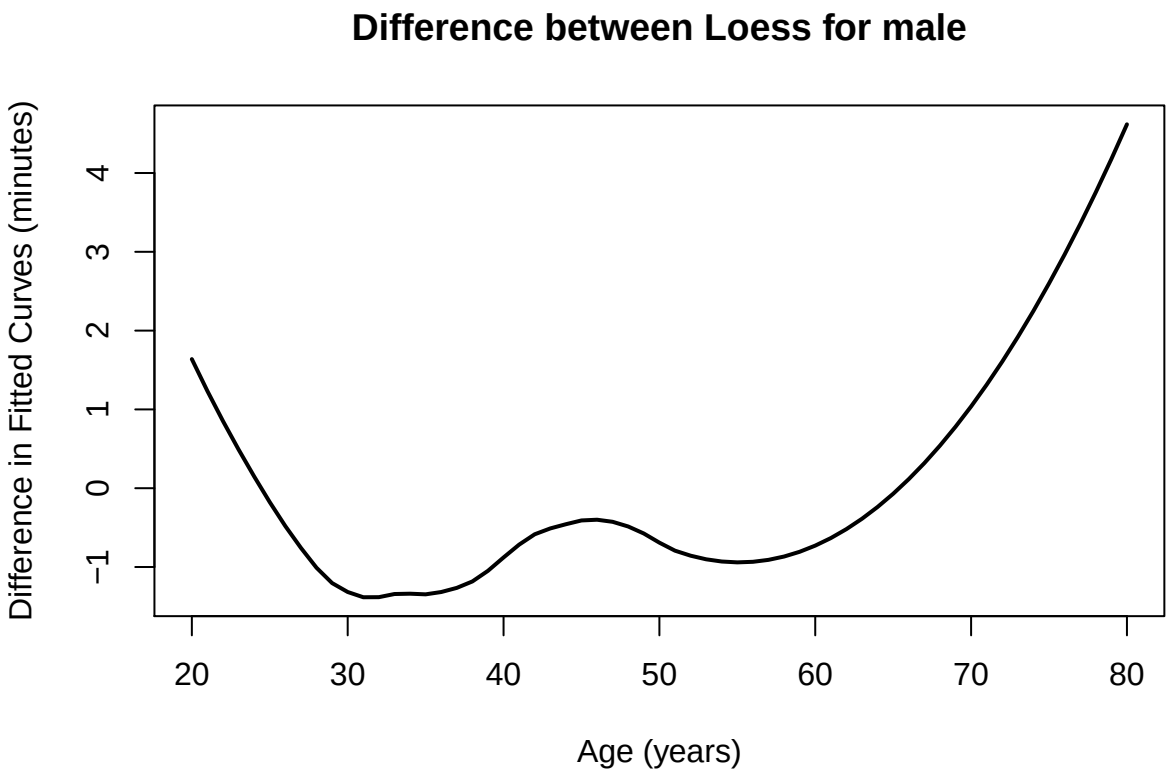

```
age2010 = cbMenSub[ cbMenSub$year == 2010, "age" ]
age2012 = cbMenSub[ cbMenSub$year == 2012, "age" ]

## Density Curves for the Age of Male Runners for 2 years
plot(density(age2010, na.rm = TRUE),
     ylim = c(0, 0.05), col = "purple",
     lwd = 3,  xlab = "Age (years)",
     main = "Density Curves for Male Runners")

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")
```




```
mR.lo01 = loess(runTime ~ age, cbMenSub[ cbMenSub$year == 2010,])
mR.lo.pr01 = predict(mR.lo01, data.frame(age = age20to80))

mR.lo12 = loess(runTime ~ age, cbMenSub[ cbMenSub$year == 2012,])
mR.lo.pr12 = predict(mR.lo12, data.frame(age = age20to80))

## Loess Curves Fit to Performance for 2 years
plot(mR.lo.pr01 ~ age20to80,
     type = "l", col = "purple", lwd = 3,
     xlab = "Age (years)", ylab = "Fitted Run Time (minutes)",
     main="Loess Fit to Performance for male")

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")
```




```
gap12 = mR.lo.pr12 - mR.lo.pr01

## Difference between Loess Curves of the predicted run time for 2 years
plot(gap12 ~ age20to80, type = "l" , xlab = "Age (years)",
     ylab = "Difference in Fitted Curves (minutes)", lwd = 2,
     main="Difference between Loess for male")
```



Comparative analysis of the performance of the male runners and female runners for the years 2010, 2011 and 2012 [Based on the plots mentioned above]

- From Age by Year plot: For female runners, majority of runners fall into age range of 28 - 40 years. For male runners, majority of runners fall into age range of 30 - 45 years.
- From density plot for the age: There are more Female runner participants around age of 30 years as compared to male runners of same age.
- From runtime vs age plot: Female runners between age range of 20-50 take more time to finish the race then male runners between same age range.
- From number of runners plot: Number of female runners are more than male runners for given years(2010,2011,2012).