# Abstract

In this analysis we are studying statistics from the Cherry Blossom Ten Mile Run from 1999 to 2012. Initially we need to scrape the website for the Cherry Blossom event to capture the runtime and age for years 1999 through 2012 for men. We adapted this code to capture the same metrics for women so that we can run comparative analysis between the two groups of runners. After we ensure that we have the statistics for both groups, we plot graphs comparing actual runtimes and predicted run times using age as the explanatory variable. In the end we find that both groups share similar patterns, but men have a average 5-15 minute advantage over women for runtime at a comparable age.

# Introduction

In 1912, the mayor of Tokyo Japan gifted to Washington D.C. 3000 cherry trees. The Cherry Blossom festival traces its origins to a tree planting reenactment held in 1927. [1] The Cherry Blossom Festival 10-mile race is held during the festival in early April each year. The first race was held in 1973 and has been held each year since then. [2] Over the years the festival and the race have grown in popularity so much that race organizers limit the racers to specific fitness levels. If you can not complete the 10-mile race in the 2 hours 20 minutes time period it is recommended you take part in the 5K instead of the 10-mile race. [3]

The official site for the Cherry Blossom race ([www.cherryblossom.org](https://smu365-my.sharepoint.com/personal/smillett_smu_edu/Documents/School/MSDS%207333%20Quantifying%20the%20World/Session%208/www.cherryblossom.org)) maintains an online copy of the official race results dating back to 1999. Only the results from 1999 to 2012 are published in their entirety and available for download. There are separate HTML pages for women’s results and for men’s results for all 14 years. The names of the HTML pages for each year do not follow any pattern so each page must be downloaded individually by name.

Once the data is cleaned exploratory analysis is performed to identify patterns or outliers in the data. The content and type of data being analyzed leads to the exploratory techniques used. Common exploratory techniques include scatter plots, box plots, residual plots and density plots. Scatter plots can reveal outliers or areas of density within the data. Box plots show quantiles of the data (min, 1Q, mean, 3Q, max) and can provide some insight into variance. Residual plots also provide insight into variance of the data. Density plots can show points where the data may cluster. Other exploratory techniques exist for specific types of data sets such as time-series data which will not be discussed in this paper.

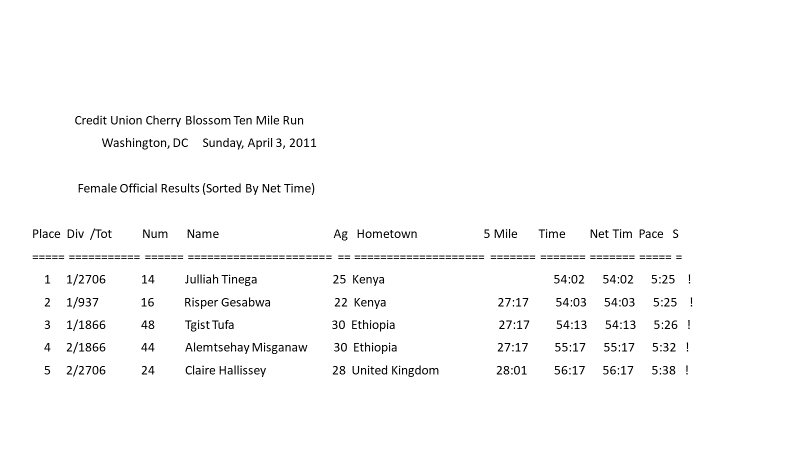
After exploratory analysis is complete and any issues of concern have been addressed, we are interested in examining the relationship between runtime and age. Scatter plots are used to determine any patterns in the data. First, simple linear regression is used to create a model describing the data. Simple linear models may not describe the data set adequately. Residual plots are examined to determine if the simple linear model is adequate, or if we need to consider a piecewise linear model or some type of non-parametric curve. Any non-linear section of the residual plot is an indicator the simple linear model is not adequate.

# Background

In this paper, we are scraping data from the internet to perform an analysis. Web scraping involves a set of techniques used find specific kinds of information, extract from the web page and aggregate into data that is unstructured and saving it in a structured database [4]. Web scraping is a topic of research in and of itself. Many research papers are available describing algorithms for web scraping. An updated collection of web scraping techniques is available in a book by Berry and Castellanos titled *Survey of Text Mining II* (2008). The book devotes an entire part to clustering methods used in web scraping.

In the case of the Cherry Blossom Festival race, the data scraped is retrieved in plain text format. An example of the data set from the men’s results in 2011 are shown in Figure 1.

Figure 1 - First 5 Rows from 2011 Men's Results



Examination of the data in each page shows there are some differences in the ways the data are formatted from one page to the next. Although the date in Figure 1 appears to be in a consistent format, examination of other pages reveals slight differences. Figure 2 shows the spacing in the file is different and some rows have also changed.

Figure 2 - First 5 rows 2000 Men's Results

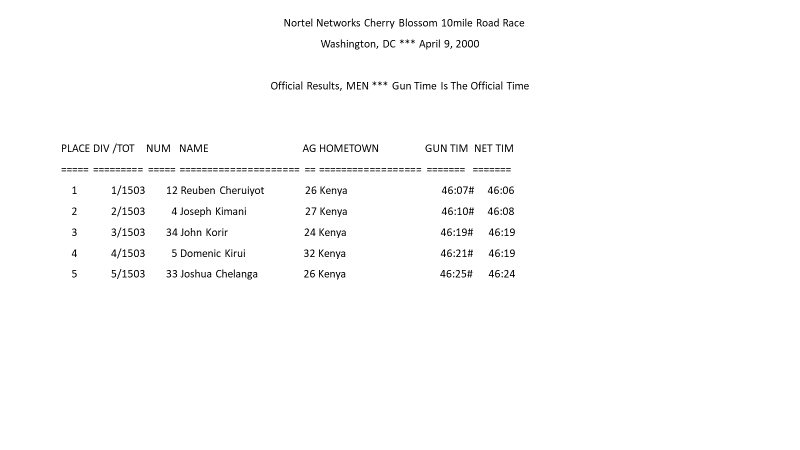
The data needs to undergo cleaning before any analysis can be done. Table 1 below describes some of the cleaning initially found by the authors in the textbook [2].

Table 1 - Partial list of cleaning required

|  |
| --- |
| Tables have headers |
| The last line of the header is a row of ‘=’ characters acting as a separation line |
| There are blanks in the row of ‘=’ characters |
| The row above the ‘=’ character contains column names |
| The 2011 file has 2 times reported and only 1 time. |

Data scraped from the internet is often “dirty”; that is, it is in an unstructured format and must be “cleaned” before it can be saved for use during analysis. Cleaning data is an iterative process. As one part of the data is formatted and put into a structured format, other areas that pose problems appear. The use of regular expressions to search and delete or modify data to fit into a structured format is a common exercise. Regular expressions make use of wildcard characters and pattern matching allowing a single line of code to structure data from different types of unstructured code.

Our structured data set is very large. When examining scatter plots with such large data sets, it is very easy for the scatter plot to be subject to over plotting. Over plotting occurs when there is so much data it is difficult to identify any patterns in the data. There are techniques available to get past this issue. We can change the symbol used in the plot from a circle to a disk and reduce the size of the symbol. Add a level of transparency to the color of the symbol has the effect of making high density observations appear darker than low density areas [2].

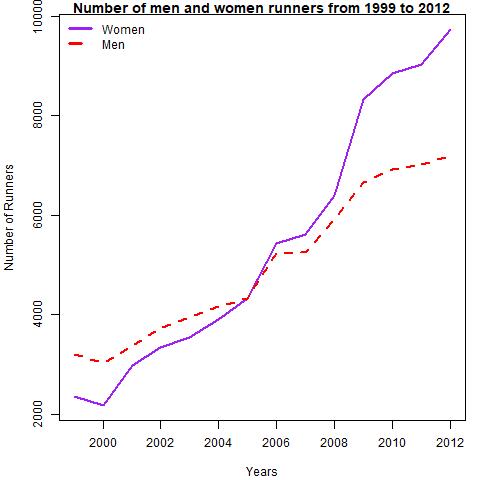
The final task is to ascertain any relationship with age and time to complete the race. We want to understand if the age and time to complete the race can be fit to a simple linear model or are the other subtleties in the data that a simple linear model can not explain. The first step is to examine the residual plot. Fitting a LOESS (Locally Weighted Smoothing) curve to the residual plot can reveal if the relationship is linear, or if a piecewise linear model or non-parametric curve better describes the data. LOESS creates a smooth line through a data set which helps see the relationship between the age and time variables. LOESS is generally used to fit a line when noise is present in the data or there are weak relationships between the data causing difficulty in identifying patterns. LOESS has the advantage being a flexible algorithm to fit data and is easy to use. It is also computationally inexpensive. The downside to LOESS is it can create models that are difficult to interpret the models created. The equations are usually not simple to understand can be very complex.

# Method

We will be taking code that was originally provided to scrape the Cherry Blossom Ten Mile Run website (www.cherryblossom.org) for the Men run times for years 1999 to 2012. There were some issues in adapting the code to scrape the data from the pages since many of the pages have different formatting from year-to-year and between the male and female times. We found that after careful analysis of problematic pages that we were able to collect all of the ages and run times for the female runners.

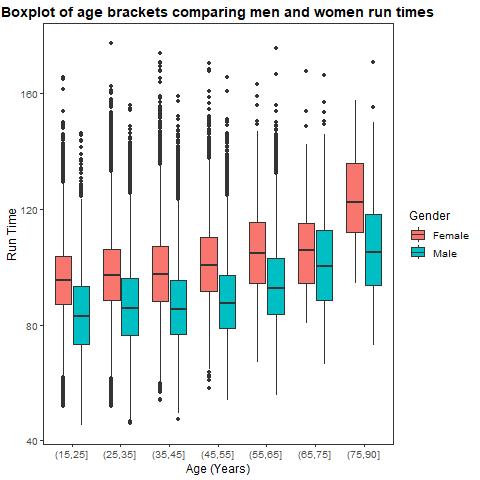
We initially wanted to compare the number of runners between both populations between men and women. In Figure 3 there is an obvious trend in increased interest in the Cherry Blossom run for both Men and Women, but in 2005 you see the interest from women eclipse that of men. There is an obvious surge in popularity among women following 2005 with only gaining popularity. It is interesting to see this change and is an area that further research could be explored.

Figure 3 Number of Runners



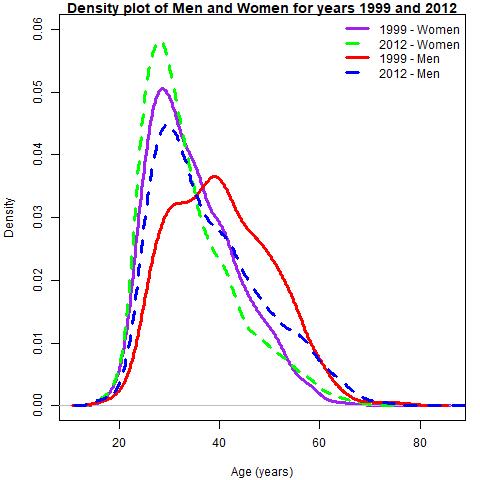
After noting the number of runners and the difference between men and women we wanted to see how their initial run times compared. We created groups of ages based on 10-year buckets and plotted them, comparing men and women run times side-by-side. In Figure 4 there is a marked difference between the runtimes between the men and women in all age brackets. In every example men have lower median runtimes. The difference is so large that in all but one age bracket, 65 to 75, the median run time of women is greater than the top 75% of observations of men. This difference in the distribution of run times between the two sexes is interesting and could warrant further research. Also breaking these categories out among different years to see if the trends have changed overtime could provide insight as to why the differences are so large.

Figure 4 Boxplot of age brackets



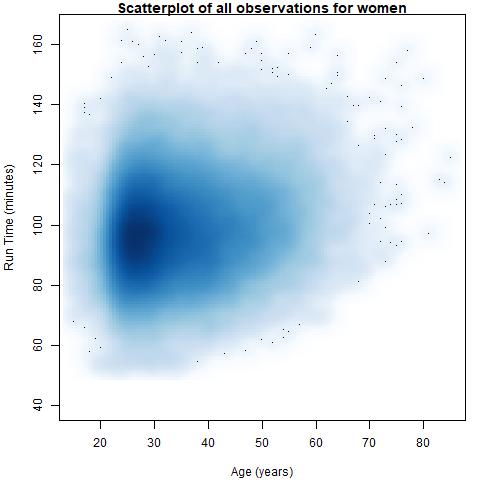
To explore the changes over the years in terms of the make-up of runners we plotted density plots of male and female runners in 1999 and 2012. Figure 5 shows an interesting shift in the age distribution for men and women. For men, there is a shift in most of the runners being over 40 in 1999 to a younger group in 2012 with an average age of about 30. This distribution mirrors what both years for women look like, with both 1999 and 2012 showing the mean at about 30. All the distributions are right skewed, with the 1999 men’s numbers showing the closest to a normal distribution.

Figure 5 Density plots between men and women



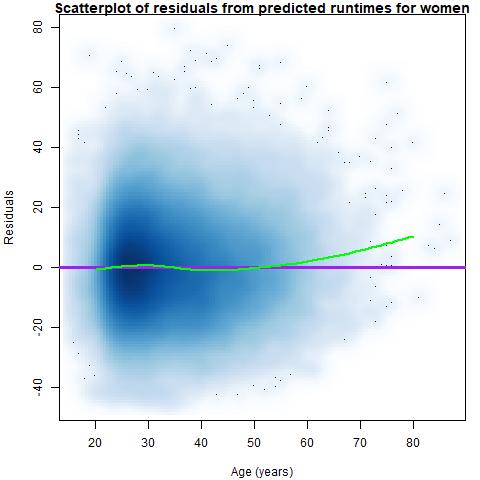
Next, we plotted all the points for women with run time being explained by age, see Figure 6. We can see how this scatterplot mimics the last two graphs. First there is an obvious right skew for the distribution of age, with the mean falling on about 30. Also, there is a nearly normal distribution of average run times with the mean falling on about 100 min and a range of about 50 to 150. Next, we will do analysis to see what the distribution of residuals are based on doing LOESS regression.

Figure 6 Scatterplot of women



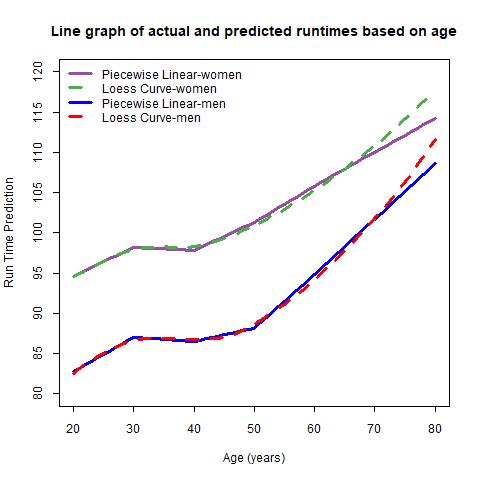
In Figure 7 we plotted the distribution of the residuals around a piecewise-linear plot for the relationship between run time and age. We see that the distribution appears uniform, but the green line shows that there is there is a lack of fit to the plot at the upper range of the X axis. The green line shows a LOESS prediction which indicates that there is a non-linear relationship between the explanatory and response variable. Next, we will plot as actual plots on the standard x-y plot of age on run time.

Figure 7 Distribution of residuals from LOESS



In Figure 8 we are showing the comparison in the results between the predicted piecewise linear runtimes for men and women, as compared to the LOESS curve. We can see similar patterns between both men and women plots where there is a larger deviation from the linear plot compared to the LOESS at the right hand of the x axis. There is evidence possibly of the deviation being larger for women, indicating that women’s runtimes have a steeper increase as they age.

Figure 8 Actual and predicted values for men and women



# Results

Comparing the results between men and women in the Cherry Blossom event shows some similarities between the two groups such as similar distribution of runners that participate in the event and they show similar performance graphs among the populations as they age. We should not infer these trends across all runners since this is a particular audience that would sign up for this event. Some differences we found between men and women are the increased popularity for female runners and the change in distribution of ages for men from 1999 until 2012. Plotting men and women together showed interesting relationships, like how men consistently outperform women.

# Future Work

The results between the men and women run times bring up a lot of additional questions for future research. It is interesting to see the increased attendance among women that are attending the event, it would be interesting to see why there is such interest from women. Is this a result because of increased advertising for the event, current trends in fitness or another factor like not being able to get into other events? Even with requirements for minimum performance times, there is an interesting trend in the popularity for the event among women. To get this information additional statistics would have to be gathered for the attendees like survey data or other information about how the event is put on.

With the data that was already collected we could do additional analysis. There is home data that was dropped for this analysis which gives information on individual runner’s country of origin or state if they are from the USA. It would be interesting to see stats on how well people perform according to their home. Using home information, it would be interesting to see the change in attendance numbers. Are more people attending the run now that live close to the area or is this growing into being a destination event. Certain marathons are attended by people all over the world like the Boston Marathon, so as the event has become more popular, which groups are driving this increase in attendance? Performance of individual racers could be tracked through the years.

# References

|  |  |
| --- | --- |
| [1] | "History of the Cherry Blossom Trees and Festival," NCBF, [Online]. Available: https://nationalcherryblossomfestival.org/about/history/. [Accessed 22 October 2018]. |
| [2] | D. N. Lang and D. T. Lang, "Chapter 2 - Modeling Runners' Times in the Cherry Blossom Race," in *Data Science in R - A Case Studies Approach to Computational Reasoning and Problem Solving*, Boca Rotan, CRC Press, 2015, p. 50. |
| [3] | "Cherry Blossom - Entry Information," [Online]. Available: http://www.cherryblossom.org/generalinfo/entryinfo.php. [Accessed 22 October 2018]. |
| [4] | U. M. Vargiu Eloisa, "Exploiting web scraping in a collaborative filtering based approach to web advertising," *Artificial Intelligence Research,* vol. 2, no. 1, pp. 44-54, 2013. |

# Appendix

R Code

library(XML)

library(changepoint)

library(ggplot2)

setwd('C:\\Users\\Steven\\Dropbox\\School\\MSDS 7333 Quantifying the World\\Session 8\\')

ubase <- 'http://www.cherryblossom.org/'

menURLs <-

c("results/1999/cb99m.html", "results/2000/Cb003m.htm", "results/2001/oof\_m.html",

"results/2002/oofm.htm", "results/2003/CB03-M.HTM",

"results/2004/men.htm", "results/2005/CB05-M.htm",

"results/2006/men.htm", "results/2007/men.htm",

"results/2008/men.htm", "results/2009/09cucb-M.htm",

"results/2010/2010cucb10m-m.htm",

"results/2011/2011cucb10m-m.htm",

"results/2012/2012cucb10m-m.htm")

femURLs <- c("results/1999/cb99f.html", "results/2000/Cb003f.htm", "results/2001/oof\_f.html",

"results/2002/ooff.htm", "results/2003/CB03-F.HTM",

"results/2004/women.htm", "results/2005/CB05-F.htm",

"results/2006/women.htm", "results/2007/women.htm",

"results/2008/women.htm", "results/2009/09cucb-F.htm",

"results/2010/2010cucb10m-f.htm",

"results/2011/2011cucb10m-f.htm",

"results/2012/2012cucb10m-f.htm")

women\_urls <- paste(ubase, femURLs, sep= '')

men\_urls <- paste(ubase, menURLs, sep= '')

# options(error = recover)

##Set GGplot to plain

theme\_set(theme\_bw())

theme\_update(text = element\_text(size=12),

panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),

strip.background = element\_blank()

)

#

# Retrieve data from web site,

# find the preformatted text,

# and write lines or return as a character vector.

#

extractResTable = function(url = "http://www.cherryblossom.org/results/2009/09cucb-F.htm",

year = 1999, sex = "female", file = NULL){

doc = htmlParse(url)

if (year == 2000) {

# Get preformatted text from 4th font element

# The top file is ill formed so the <pre> search doesn't work.

ff = getNodeSet(doc, "//font")

txt = xmlValue(ff[[4]])

els = strsplit(txt, "\r\n")[[1]]

}

else if (year == 1999) {

# Get preformatted text from 4th font element

# The top file is ill formed so the <pre> search doesn't work.

ff = getNodeSet(doc, "//pre")

txt = xmlValue(ff[[1]])

els = strsplit(txt, "\n")[[1]]

}

else if (year == 2009 & sex == "male") {

# Get preformatted text from <div class="Section1"> element

# Each line of results is in a <pre> element

div1 = getNodeSet(doc, "//div[@class='Section1']")

pres = getNodeSet(div1[[1]], "//pre")

els = sapply(pres, xmlValue)

}

else {

# Get preformatted text from <pre> elements

pres = getNodeSet(doc, "//pre")

txt = xmlValue(pres[[1]])

els = strsplit(txt, "\r\n")[[1]]

}

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

# Write the lines as a text file.

writeLines(els, con = file)

}

years <- 1999:2012

femTables <- mapply(extractResTable, url = women\_urls, year = years)

menTables <- mapply(extractResTable, url = men\_urls, sex = 'male', year = years)

names(femTables) = years

names(menTables) = years

sapply(femTables, length)

save(femTables, file = "Data\CBFemaleTextTables.rda")

save(menTables, file = "Data\CBMaleTextTables.rda")

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(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])

},

headerRow = headerRow, searchLocs = searchLocs )

}

extractVariables = function(file, backup\_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]

if (spacerRow==" ")

{

alt\_eqIndex = grep("^===", backup\_file)

spacerRow = backup\_file[alt\_eqIndex]

# Extract the two key rows and the data

headerRow = tolower(backup\_file[ alt\_eqIndex - 1 ])

}

##Checks to see if t

else

{

headerRow = tolower(file[ eqIndex - 1 ])

}

body = file[ -(1 : eqIndex) ]

# 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)

}

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)

}

###Conditional block to handle 2006 data

separatorIdx = grep("^===", femTables[["2006"]])

separatorRow = femTables[['2006']][separatorIdx]

separatorRowX = paste(substring(separatorRow, 1, 63), " ",

substring(separatorRow, 65, nchar(separatorRow)),

sep = "")

femTables[['2006']][separatorIdx] = separatorRowX

#extracting data to womenResMat

names(femTables) <- 1999:2012

womenResMat <- mapply(extractVariables, femTables,menTables)

womenDF = mapply(createDF, womenResMat, year = 1999:2012,

sex = rep("W", 14), SIMPLIFY = FALSE)

cbWomen = do.call(rbind, womenDF)

cbWomen = cbWomen[complete.cases(cbWomen),]

save(cbWomen,file= 'Data\cbWomen.rda')

#extracting data to womenResMat

menResMat <- mapply(extractVariables, menTables,menTables)

menDF = mapply(createDF, menResMat, year = 1999:2012,

sex = rep("M", 14), SIMPLIFY = FALSE)

sapply(menDF, function(x) sum(is.na(x$runTime)))

separatorIdx = grep("^===", menTables[["2006"]])

separatorRow = menTables[['2006']][separatorIdx]

separatorRowX = paste(substring(separatorRow, 1, 63), " ",

substring(separatorRow, 65, nchar(separatorRow)),

sep = "")

menTables[['2006']][separatorIdx] = separatorRowX

menResMat <- mapply(extractVariables, menTables,menTables)

menDF = mapply(createDF, menResMat, year = 1999:2012,

sex = rep("M", 14), SIMPLIFY = FALSE)

sapply(menDF, function(x) sum(is.na(x$runTime)))

cbMen = do.call(rbind, menDF)

cbMen = cbMen[complete.cases(cbMen),]

save(cbMen,file= 'Data\cbMen.rda')

jpeg("Output/CB\_SmoothScatter.jpeg")

oldPar = par(mar = c(4.1, 4.1, 1, 1))

smoothScatter(y = cbWomen$runTime, x = cbWomen$age,

ylim = c(40, 165), xlim = c(15, 85),

xlab = "Age (years)", ylab = "Run Time (minutes)")

title('Scatterplot of all observations for women')

par(oldPar)

dev.off()

cbWomenSub = cbWomen[cbWomen$runTime > 30 &

!is.na(cbWomen$age) & cbWomen$age > 15, ]

cbMenSub = cbMen[cbMen$runTime > 30 &

!is.na(cbMen$age) & cbMen$age > 15, ]

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

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

table(ageCat)

#Plot stratified boxplots

jpeg("Output/CB\_Boxplots.jpeg")

oldPar = par(mar = c(4.1, 4.1, 1, 1))

tempdata <- data.frame("runTime"=cbWomenSub$runTime,"Age"= ageCat\_women,"Gender"="Female")

tempdata2 <- data.frame("runTime"=cbMenSub$runTime,"Age"= ageCat\_men,"Gender"="Male")

tempdata <- rbind(tempdata,tempdata2)

# boxplot(runTime~interaction(Age, Gender, lex.order = T),data=tempdata,

# xlab = "Age (years)", ylab = "Run Time (minutes)")

ggplot(tempdata, aes(x=as.factor(Age),y=runTime,fill=Gender))+geom\_boxplot()+

xlab('Age (Years)')+ylab('Run Time')+

ggtitle('Boxplot of age brackets comparing men and women run times') +

theme(plot.title = element\_text(hjust = 0.5,face='bold'))

par(oldPar)

dev.off()

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

lmAge$coefficients

summary(lmAge)

class(lmAge)

jpeg("Output/CB\_ResidSimpleLM.jpeg")

oldPar = par(mar = c(4.1, 4.1, 1, 1))

smoothScatter(x = cbWomenSub$age, y = lmAge$residuals,

xlab = "Age (years)", ylab = "Residuals")

abline(h = 0, col = "purple", lwd = 3)

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)

title('Scatterplot of residuals from predicted runtimes for women')

par(oldPar)

dev.off()

womenRes.lo = loess(runTime ~ age, cbWomenSub)

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

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

menRes.lo.pr = predict(menRes.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\_women = lapply(decades,

function(x) pmax(0, (cbWomenSub$age - x)))

names(overAge\_women) = paste("over", decades, sep = "")

overAge\_women = as.data.frame(overAge\_women)

# tail(overAge)

lmPiecewise\_women = lm(runTime ~ . ,

data = cbind(cbWomenSub[, c("runTime", "age")],

overAge\_women))

overAge\_men = lapply(decades,

function(x) pmax(0, (cbMenSub$age - x)))

names(overAge\_men) = paste("over", decades, sep = "")

overAge\_men = as.data.frame(overAge\_men)

lmPiecewise\_men = lm(runTime ~ . ,

data = cbind(cbMenSub[, c("runTime", "age")],

overAge\_men))

# 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)

# tail(overAgeDF)

predPiecewise\_women = predict(lmPiecewise\_women, overAgeDF)

predPiecewise\_men = predict(lmPiecewise\_men, overAgeDF)

jpeg("Output/CB\_PiecewiseLoessCurves.jpeg")

plot(predPiecewise\_women ~ age20to80,

type = "l", col = "#984ea3", lwd = 3,

axes=FALSE,

ylim = c(80, 120),

xlab = "Age (years)", ylab = "Run Time Prediction")

axis(side =1 , tck=-.015,at = c(seq(from=20,80,10)))

axis(side =2 , tck=-.015,at = c(seq(from=80,120,5)))

box()

lines(x = age20to80, y = womenRes.lo.pr, col = "#4daf4a", lwd = 3, lty = 2)

lines(predPiecewise\_men ~ age20to80,

type = "l", col = "blue", lwd = 3,

# type = "l", col = "purple", lwd = 2,

xlab = "Age (years)", ylab = "Run Time Prediction")

lines(x = age20to80, y = menRes.lo.pr, col = "red", lwd = 3, lty = 2)

legend("topleft", col = c("#984ea3", "#4daf4a", 'blue','red'), lty = c(1, 2), lwd = 3,

legend = c("Piecewise Linear-women", "Loess Curve-women","Piecewise Linear-men", "Loess Curve-men"),

bty = "n")

title('Line graph of actual and predicted runtimes based on age')

dev.off()

jpeg("Output/CB\_NumRunnersLinePlot.jpg")

oldPar = par(mar = c(4.1, 4.1, 1, 1))

numRunners\_women = with(cbWomen, tapply(runTime, year, length))

numRunners\_men = with(cbMen, tapply(runTime, year, length))

plot(numRunners\_women ~ names(numRunners\_women), type="l", lwd = 2, col='purple',

xlab = "Years", ylab = "Number of Runners")

lines(numRunners\_men ~ names(numRunners\_men), type="l", lwd = 2, lty=2, col='red')

legend("topleft", col = c("purple", "red"), lty = c(1, 2), lwd = 3,

legend = c("Women", "Men"),

bty = "n")

title('Number of men and women runners from 1999 to 2012')

par(oldPar)

dev.off()

jpeg("Output/CB\_AgeDensity99vs12.jpg")

oldPar = par(mar = c(4.1, 4.1, 1, 1))

age1999 = cbWomenSub[ cbWomenSub$year == 1999, "age" ]

age2012 = cbWomenSub[ cbWomenSub$year == 2012, "age" ]

men\_age1999 = cbMenSub[ cbMenSub$year == 1999, "age" ]

men\_age2012 = cbMenSub[ cbMenSub$year == 2012, "age" ]

plot(density(age1999, na.rm = TRUE),

ylim = c(0, 0.06), col = "purple",

lwd = 3, xlab = "Age (years)", main = "")

title('Density plot of Men and Women for years 1999 and 2012')

lines(density(age2012, na.rm = TRUE),

lwd = 3, lty = 2, col="green")

lines(density(men\_age1999, na.rm = TRUE),

lwd = 3, lty = 1, col="red")

lines(density(men\_age2012, na.rm = TRUE),

lwd = 3, lty = 2, col="blue")

legend("topright", col = c("purple","green",'red','blue'), lty= 1:2, lwd = 3,

legend = c("1999 - Women","2012 - Women",'1999 - Men','2012 - Men'), bty = "n")

par(oldPar)

dev.off()