Modeling Runners’ Times in the Cherry Blossom Race

Vinh Le and Marie Wallmark

### Southern Methodist University, Masters in Data Science, Dallas, TX

Introduction

The Cherry Blossom Ten Mile Run is a yearly road race held in Washinton D.C. in early April. The participants include both male and female runners ranging in age from 9 to 89. Following the race, the results are published for public access on the organizer’s website. The data can be scrapped from the website for analysis. For each year, data is provided for the runner’s place in the race, division, number, name, age, hometown, 5-mile time, final time, and pace. The data for the most part maintains this format except for some deviations. In addition to our statistical analysis of the data, we also undertake the task of formatting this data in such a way as to be analyzed. Though the race started in 1973, the earliest results posted are for 1999. Our analysis into this relationship looks at the 14-year time span between 1999 and 2012. This data allows for insight into the relationship between age and performance.

We start our investigation looking at the men’s data, and then later in our analysis we turn to the women’s data to examine age distribution across the same 14-year time span. Finally, we conclude our analysis with an overall comparison of age distribution between male and female runners.

Background

The primary interest of this analysis is to understanding how people’s physical performance changes as they age. More specifically, we are trying to detect change over a period of time. The problem of estimating the point at which the statistical properties of a sequence of observation changes, is referred to as change point detection. The change can either be in means or variance, and may occur as a single changepoint or multiple changepoints. Accurate representation of the data requires that we be able to determine how many segments are needed (how many changepoints are present), and estimate the values of the parameters associated with each segment [2]. A test statistics such as the likelihood ratio can be constructed to determine if a change has occurred, and can be extended to multiple changes by summing the likelihood for each of the m segments [2]. Using the likelihood ratio method “requires the calculation of the maximum log-likelihood under both the null and alternative hypothesis” [2]. The null hypothesis is reject based on the chosen threshold value. There remains great challenge in determining the appropriate value for this threshold.

Methods

The Cherry Blossom Ten Mile Run can be scraped from the web to be read in R. The race results are stored in tables named 1999 to 2012 available at http://www.cherryblossom.org. Our task is to download them from the CherryBlossom website and extract them into text files that include tables for 1999.txt, …, 2012.txt. Through examination, we can see that the tables are formatted differently for 1999, 2000, and 2009. The menURLs format were also different between the years. To fix that we manually insert all the correct links in the variable menURLs.

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")

When using the original extractResTable function it gave us a length of 1 for 1999, 2000, and 2009. This created another problem because it means that those tables were not downloaded correctly. To correct that we had to modify the extractResTable function to include a condition that take in effect of the different format for 1999, 2000, and 2009. The modified extractResTable function is included in the R notebook.

Once we got the modified function working, we created a variable fileNames which included the name of the files we want to save in our directory. The next step is then to call the function with the parameter fileNames included. Once we have all 1999.txt, …, 2012.txt files we then put them in a directory MenTxt. The whole process was then repeated for women and then saved in WomenTxt folder.

We are interested in understanding how people’s performance change with age. To begin the analysis, we begin with an initial examination of the 2012 table to get a sense for the structure of the data. From there, we create functions that can be generalized to extract the data from the other years in a format that can be analyzed.

#m2012 <- read.table(file = "MenTxt/2012.txt", skip = 8)  
els = readLines("MenTxt/2012.txt")  
els[1:10]

The extractVariables function incorporates the two functions (findColLocs and selectCols) used earlier to locate the columns and applies this to each year, resulting in dataframes for each year.

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

Here the initial variables of interest are the runner’s name, hometown, age, and the three versions of time (gun, net, and plain time). We next convert variables into appropriate data types for statistical analysis. In the process of converting data types, we discover some error in the formatting of the data that resulted in erroneous numbers for the age variable in 2003.

age = as.numeric(menResMat[['2012']][ , 'ag'])  
  
tail(age)

## [1] 41 39 56 35 NA 48

The selectCols function is revised to change the index for the end of each variable when the extraction is performed. This should correct the position of age to reflect the actual age of the runners.

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 )  
}

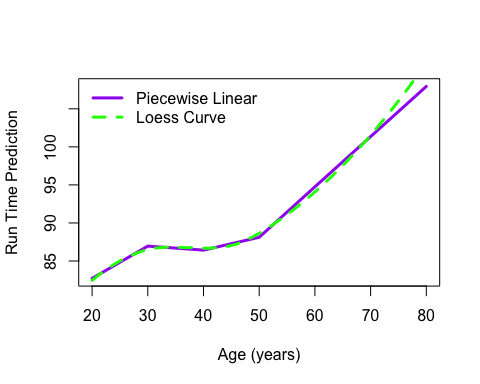
We created age variables and corrected ages that was imported incorrectly. Then we convert time variables into charTime to show the correct time that the runner took to run finish the race.

The next step is to create various scatter plots to see the distribution of all male runners who completed the race, improving plot to avoid over fitting

fitting model to average performance

lmAge = lm(runTime ~ age, data = cbMenSub)  
  
lmAge$coefficients

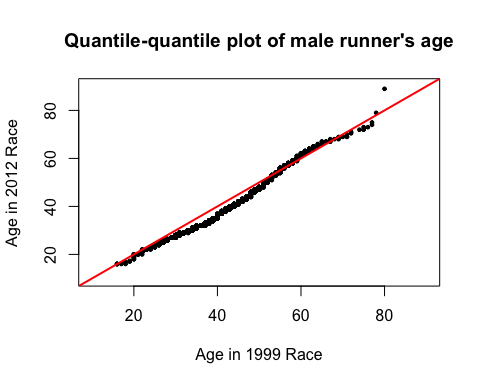
## (Intercept) age   
## 78.7567186 0.2252921

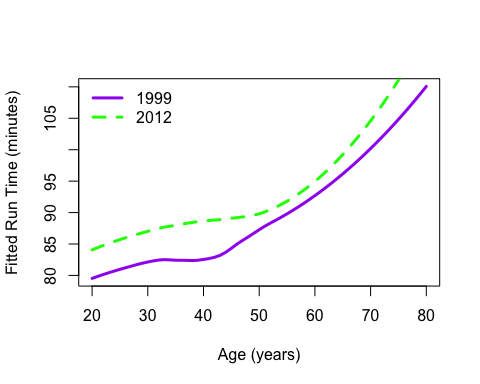


The above figure is a Linear and Loess Curves Fitted to Run Time vs Age. You can see that the Linear and Loess curve are very similar as they laid right on top of each other. The only exception is between age 70 and 80 years old.

Above is the Loess Curves Fit to Performance for all 1999 through 2012 Male Runners between the age 20 and 80 years old. Both curves have similar shape and are almost right on top of each other. The exception is between 70 and 80 years old.

cross sectional comparison: seeing how runners performance changes as they age. difference in predicted run times for the two groups (1999 and 2012)

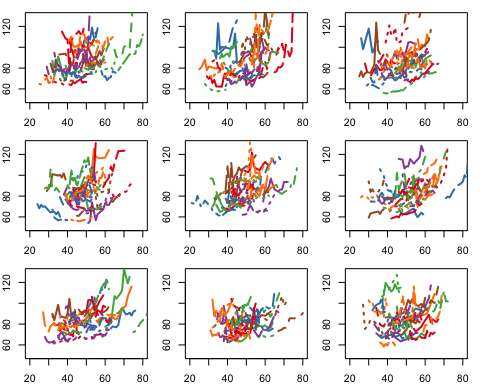




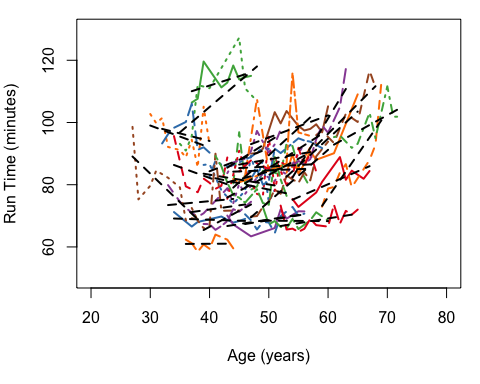
Above is a Loess Curves Fit o Perfomarnce for 1999 and 2012 Male Runners. The run time for male runners in 2012 sit above the 1999 male runners. The gap between 1999 and 2012 is about 5 minutes. The only exception is between 50 and 60 years old with only about 2 to 3 minutes.

comparison of the run time age relationship for all 14 years

modeling change in run times for individuals



This is the run time from multiple races. These plots show the times for male runners who completed at least 8 races. Each set of connected segments corresponds with the run time for one athlete. All the plots have similar shape but the main thing to point out is that they show an upward curve with age.



This is a plot of Linear Fits of Run Time to Age for Individual Runners. This is where we augmented the bottom-right plot from the previous steps with the least squares fit of run time for each athlete.

Question 10

Using qq-plots, boxplots, and density curves to compare age distribution of both male and female runners across all 14 years

library(tidyverse)

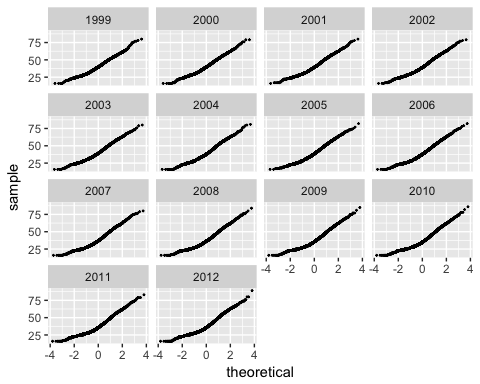
## Loading tidyverse: ggplot2  
## Loading tidyverse: tibble  
## Loading tidyverse: tidyr  
## Loading tidyverse: readr  
## Loading tidyverse: purrr  
## Loading tidyverse: dplyr

## Conflicts with tidy packages ----------------------------------------------

## filter(): dplyr, stats  
## lag(): dplyr, stats

#MEN  
# QQ plots  
ggplot(cbMenSub, aes(sample = age)) +  
 geom\_point(stat = "qq", size = .25) +  
 facet\_wrap(~as.factor(year))

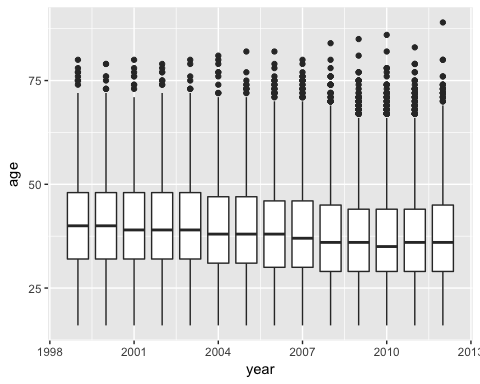
Looking at the age distribution of male runners across all 14 years



ggsave("CB\_qq99thru12.pdf", dpi = 300)

## Saving 5 x 4 in image

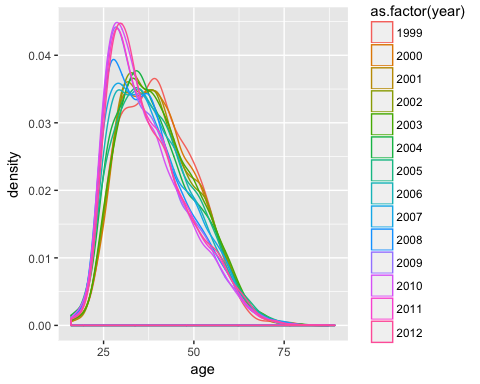
# Boxplots  
ggplot(cbMenSub, aes(year, age, group = as.factor(year))) +  
 geom\_boxplot()



ggsave("CB\_box99thru12.pdf", dpi = 300)

## Saving 5 x 4 in image

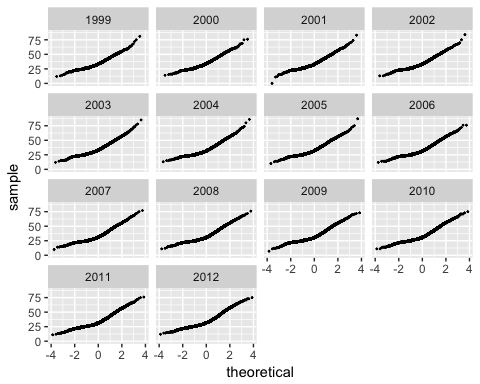
# Density curves  
ggplot(cbMenSub, aes(age, colour = as.factor(year))) +  
 geom\_density()



ggsave("CB\_density99thru12.pdf", dpi = 300)

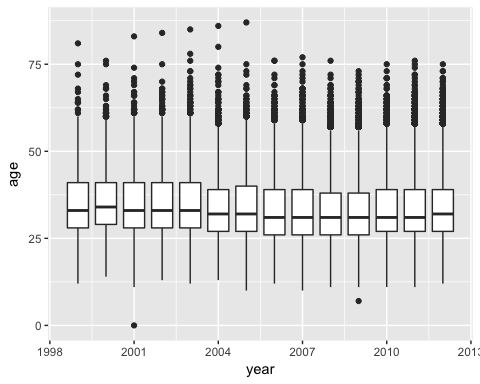
Looking at age distribution for female runners

#WOMEN  
# QQ plots  
ggplot(cbWomen, aes(sample = age)) +  
 geom\_point(stat = "qq", size = .25) +  
 facet\_wrap(~as.factor(year))



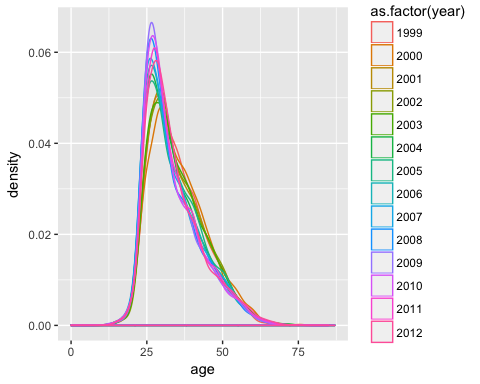
ggsave("CB\_qq99thru12\_women.pdf", dpi = 300)

# Boxplots  
ggplot(cbWomen, aes(year, age, group = as.factor(year))) +  
 geom\_boxplot()



ggsave("CB\_box99thru12\_women.pdf", dpi = 300)

ggplot(cbWomen, aes(age, colour = as.factor(year))) +  
 geom\_density()



ggsave("CB\_density99thru12\_women.pdf", dpi = 300)

# Men: median ages by year  
cbMen %>% group\_by(year) %>% filter(!is.na(age)) %>% summarise(med = median(age), dev = sd(age))

## # A tibble: 14 x 3  
## year med dev  
## <int> <dbl> <dbl>  
## 1 1999 40 10.27307  
## 2 2000 40 10.51276  
## 3 2001 39 10.65323  
## 4 2002 39 10.70625  
## 5 2003 39 10.77123  
## 6 2004 38 10.94539  
## 7 2005 38 11.05366  
## 8 2006 38 10.94984  
## 9 2007 37 11.14528  
## 10 2008 36 10.90176  
## 11 2009 35 10.79945  
## 12 2010 35 10.76670  
## 13 2011 36 10.84948  
## 14 2012 35 10.87064

# Women: median ages by year  
cbWomen %>% group\_by(year) %>% filter(!is.na(age)) %>% summarise(med = median(age), dev = sd(age))

## # A tibble: 14 x 3  
## year med dev  
## <int> <dbl> <dbl>  
## 1 1999 33 8.926514  
## 2 2000 34 9.319658  
## 3 2001 33 9.201297  
## 4 2002 33 9.231994  
## 5 2003 33 9.372920  
## 6 2004 32 9.287319  
## 7 2005 32 9.387782  
## 8 2006 31 9.254709  
## 9 2007 31 9.284784  
## 10 2008 31 9.116972  
## 11 2009 31 8.981665  
## 12 2010 31 9.091657  
## 13 2011 31 9.212487  
## 14 2012 32 9.274854

Results

Age of men shows greater dispersion over the thirteen years of the race, and a stronger chronological trend towards younger runners, as shown in the summary tables. The distribution of women’s ages has not changed substantially over the years, and are tightly clustered around the median age. Additionally, men are slightly older than women on average (men median = 37, women median = 32).

Conclusion

A key take away from this analysis is the importance of ensuring the accuracy of the data. Before the analysis could be done, the data needed to be formatted and cleaned. This was a considerable endeavor.

## References

[1] D. Nolan and D. T. Lang, Data Science in R: A Case Studies Approach to Computational Reasoning and Problem Solving. Boca Raton, FL: CRC Press Taylor & Francis Group, 2015, pp.45-100.

[2] Killick, R., and Eckley, I. (2014), "changepoint: An R Package for Changepoint Analysis." Journal of Statistical Software, 58(3).