MSDS 7333 QTW - Unit8CaseStudy

Armand Post, Mustafa Sakarwala, Andrew Abbott

March 04, 2018

## Abstract:

In this case study, we are using race results for the Cherry Blossom Ten Mile Run from 1999 to 2012 to study how runners physical performance changes with age. The focus of this study involves web scraping for free and publicly published data and then getting the data cleaned into the correct format. In order to investigate how age distributions change over the years, age distribution for all runners across 14 years is compared; however, what information is recorded and how the data is formatted changes each year so close attention is paid first to formatting the data. In order for most data analysis to be viable, the data set in question usually must be cleaned and formatted. This point is especially clear in this case where web scraped data that generally comes in the form of raw character strings. The data variables were systematically extracted from the raw text files in an iterative process as new issues presented themselves. Once cleaned, the data is combined into a dataframe that can easily be used in further analysis. The R language is leveraged to appropriately format the data via statistical analysis and examining summary statistics and plots. Following data acquisition and successfully reading tables of race results into R, results are analyzed and compared using density curves, quantile-quantile plots, and boxplots. In this way, we are able to visualize the data for tens of thousands of observations to explore the performance-age relationship and ultimately examine age distribution. Furthermore, we peformed simple linear, LOESS, and piecewise regressions to more deeply explore the relationship between age and running time.

## Introduction:

Data cleaning is a very important step in the data science process. In the modern age of big data, it is very rare to find a dataset that is in a state that is conducive to effective analysis without any pre-work. It is not uncommon for a raw data set to have missing values, formatting errors, disparate layouts, and a host of other issues that need to be dealt with. Web scraping can introduce even more issues that need to be cleaned before a data set can be worked with. When scraping data from a website, it is not necessarily returned in an easy to work with format such as a table or JSON array. The data could simply be a large string of characters.

This case study was undertaken in order to explore and relationship between a person’s age and physical performance as a runner. Data used comes from the popular Credit Union Cherry Blossom race. The Cherry Blossom Ten Mile Run is held every year during the month of April in Washington, D.C., when cherry blossom trees are said to bloom. Race results for the Cherry Blossom Ten Mile Run are available to the public online at <http://www.cherryblossom.org/> and currently include annual results spanning from 1999 to 2017 (19 years in total). However for our exercise, we will focus on 1999 to 2012 since the results for those years is stored in formats that are easy to scrape with R. The race began in 1973 and has since grown by tens of thousands of runners to 17,000 entries in 2012 with ages ranging from 9 to 89. In fact, the race has been in such high demand that runners are chosen via a lottery system to enter the race. Because of the enormous amount of observations there is a clear abundance of information that can be used as a resource for investigating the performance-age relationship.

In Chapter 2 of Data Science in R: A Case Studies Approach to Computational Reasoning and Problem Solving” (Nolan and Lang) the men’s Cherry Blossom Race results for these same years is cleaned. The objective of this case study is to answer Question 2.10 in that chapter.

To that end, while this information is free and available to the public, what information is reported and how the data is formatted changes each year, presenting a challenge that must first be overcome before conducting further analysis. After data cleanup is performed and race results data tables are read into R, we then turn our focus to the age distribution of runners across all 14 years of the races, we noticed that the 1999 runners were typically older than the 2012 runners. This is achieved by creating visual representations of the data through density curves, quantile-quantile plots, and boxplots. These visualizations are important for gaining deeper insights by seeing how fast or gradual changes are over time, as well as any outliers or subtle trends. As mentioned before, a number of regressions are then run to do a deeper dive into the relationship between age and run times.

## Literature Review

To better understand web scraping, we searched for academic papers on the topic. An entry in the Journal of Statistical Software was found where a student from the Univeristy of Milan reviewed a book on web scraping entitled *Automated Data Collection with R – A Practical Guide to Web Scraping and Text Mining*. The review of the book reveals how deep the topic of web scraping can be. The book is 400 pages spanning 14 chapters. While the 2nd half of the book is focused on advanced topics such as scripting, batch processing, and statistical analysis, the first half of the book has plenty of material to cover. When scraping data from the web, one can benefit from having a breadth of knowledge including XML, JSON, APIs, HTTP, Javascript, Ajax, and also SQL to store data once retrieved. The author Iacus, further describes how workflows can break down over time due to changes in APIs. Based on the author’s review, we drew the conclusion that web scraping is a very dynamic, deep field that requires a significant investment to master.

After web scraping was addressed, we sought academic literature regarding how performance as a runner changes with age. German researchers from the University of Cologne and the Central Institute of the Federal Armed forces assessed run times for 300,757 runners. The results of their analysis was surprising. They found little evidenced of age related losses in running performance from ages 20 to 49. After the age of 50, age related performance deterioration was found, but only in the range of 2.6% to 4.4% per decade. This indicated that older athletes are able to maintain a high degree of physical plasticity. The German researchers conclusions were that performance of marathon and half marathon runners at the amateur level is more influenced by lifestyle choices. According to the www.theactivetimes.com [5], peak performance for swimmers and sprinters are at age 21 and 26 respectively. However for distance runners where endurance is more important, peak performance age increases along with distances. For marathons (26 mile run), men peak at age 29 while women peak at 30. For Iron Mans (2.4 mile swim, 112 mile bikeride, and 26 mile run within 24 hours), mean peak at 31 while women peak at age 36. The peak ages get more interesting when assesssing Ultra Events such as Triple Iron Mans (three full Iron Mans within 3 days) and Deca Iron Mans (10 full Iron Mans within 10 days). The average age of a Triple Iron Man finisher is 38 while a Deca Iron Man is an astonishing 41 years old! While the average 20 year old should be able to beat an average 50+ year old in a foot race, the younger competitor would be in for a surprise if going up against an older runner who has conditioned themselves, avoided injury, and made good lifestyle decisions over a number of decades.

First we experiment with accessing the 2012 men’s results. The first three lines are viewed and we can see that we have successfully imported the data.

## [1] ""   
## [2] " Credit Union Cherry Blossom Ten Mile Run"  
## [3] " Washington, DC Sunday, April 1, 2012"

Next we attempted to create a function to scrape data for all 14 years we are interested in assessing. The function is shown below.

Let’s test our extraction function on the 2012 data.

## [1] ""   
## [2] " Credit Union Cherry Blossom Ten Mile Run"   
## [3] " Washington, DC Sunday, April 1, 2012"   
## [4] ""   
## [5] " Official Male Results (Sorted By Net Time)"   
## [6] ""   
## [7] "Place Div /Tot Num Name Ag Hometown 5 Mile Time Pace "  
## [8] "===== =========== ====== ====================== == ==================== ======= ======= ===== "  
## [9] " 1 1/347 9 Allan Kiprono 22 Kenya 22:32 45:15 4:32 "  
## [10] " 2 2/347 11 Lani Kiplagat 23 Kenya 22:38 46:28 4:39 "  
## [11] " 3 1/1093 31 John Korir 36 Kenya 23:20 47:33 4:46 "  
## [12] " 4 1/1457 15 Ian Burrell 27 Tucson AZ 23:50 47:34 4:46 "  
## [13] " 5 3/347 19 Jesse Cherry 24 Blowing Rock NC 23:50 47:40 4:46 "

The function works! Next we created a list of all of the URls for the male racers.

## [1] "http://www.cherryblossom.org/results/1999/cb99m.html"   
## [2] "http://www.cherryblossom.org/results/2000/Cb003m.htm"   
## [3] "http://www.cherryblossom.org/results/2001/oof\_m.html"   
## [4] "http://www.cherryblossom.org/results/2002/oofm.htm"   
## [5] "http://www.cherryblossom.org/results/2003/CB03-M.HTM"   
## [6] "http://www.cherryblossom.org/results/2004/men.htm"   
## [7] "http://www.cherryblossom.org/results/2005/CB05-M.htm"   
## [8] "http://www.cherryblossom.org/results/2006/men.htm"   
## [9] "http://www.cherryblossom.org/results/2007/men.htm"   
## [10] "http://www.cherryblossom.org/results/2008/men.htm"   
## [11] "http://www.cherryblossom.org/results/2009/09cucb-M.htm"   
## [12] "http://www.cherryblossom.org/results/2010/2010cucb10m-m.htm"  
## [13] "http://www.cherryblossom.org/results/2011/2011cucb10m-m.htm"  
## [14] "http://www.cherryblossom.org/results/2012/2012cucb10m-m.htm"

Likewise, a list was created for the women.

## [1] "http://www.cherryblossom.org/results/1999/cb99f.html"   
## [2] "http://www.cherryblossom.org/results/2000/Cb003f.htm"   
## [3] "http://www.cherryblossom.org/results/2001/oof\_f.html"   
## [4] "http://www.cherryblossom.org/results/2002/ooff.htm"   
## [5] "http://www.cherryblossom.org/results/2003/CB03-F.HTM"   
## [6] "http://www.cherryblossom.org/results/2004/women.htm"   
## [7] "http://www.cherryblossom.org/results/2005/CB05-F.htm"   
## [8] "http://www.cherryblossom.org/results/2006/women.htm"   
## [9] "http://www.cherryblossom.org/results/2007/women.htm"   
## [10] "http://www.cherryblossom.org/results/2008/women.htm"   
## [11] "http://www.cherryblossom.org/results/2009/09cucb-F.htm"   
## [12] "http://www.cherryblossom.org/results/2010/2010cucb10m-f.htm"  
## [13] "http://www.cherryblossom.org/results/2011/2011cucb10m-f.htm"  
## [14] "http://www.cherryblossom.org/results/2012/2012cucb10m-f.htm"

Next we created tables for every year, we looked at the lengths for the men tables. The years 1999, 2000, and 1999 have lengths of one. The extractResTable function didn’t work properly for all years. We’ll need another approach for those that didn’t pull in correctly.

## [1] "Mens Record Numbers by Year"

## 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012   
## 1 1 3627 3727 3951 4164 4335 5245 5283 5913 1 6919 7019 7201

A new function is created with a different methodlogy for the year 2000. The new function is applied and we again assess the number of records in each dataset. The year 2000 appears to have been correctly pulled in this time.

## [1] "Mens Record Numbers by Year"

## 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012   
## 1 3019 3627 3727 3951 4164 4335 5245 5283 5913 1 6919 7019 7201

At this point, we have all years pulling correctly except 1999 and 2009. The function is further refined to correctly pull the 2009 data.

## [1] "Mens Record Numbers by Year"

## 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012   
## 1 3019 3627 3727 3951 4164 4335 5245 5283 5913 6659 6919 7019 7201

Now, every year is pulling correctly, except for 1999. After some searching on the web, the following approach was found at the github repo listed below. We test to see if it still works.

<https://github.com/bthomasbailey/cherry-blossom-run/blob/master/scrapeWeb.R>

## Loading required package: stringr

## [1] "Mens Record Numbers by Year"

## 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012   
## 1 3019 3627 3727 3951 4164 4335 5245 5283 5913 6659 6919 7019 7201

The code works, but still has issues with 1999. Next we tested on the women’s records.

## [1] "Womens Record Numbers by Year"

## 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012   
## 1 2169 2976 3338 3547 3907 4342 5445 5699 6405 8333 8863 9038 9738

We have the same issues as with the men. All years appear to have pulled correctly, except for 1999. We proceeded to save our tables to local machines and then read them back in to see if that might help with 1999 reading as one long text string.

To begin converting to dataframes, we save all of the data in .rda files.

Next we create some helper functions to read back in the text files we saved previously.

After the helper functions are created, the text files are read into the R session.

The text files were correctly read in, so the next step is to apply our function to extract the variables. Notice here that we include a new column that represents the gender of the racer so that later we can combine dataframes to compare the numbers for the men and women.

Let’s check the number of records again.

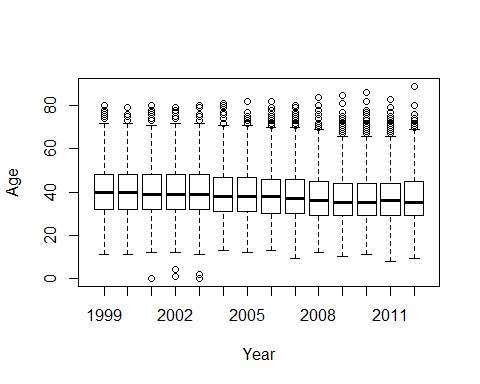
## [1] "Mens Record Numbers by Year"

## 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012   
## 3190 3016 3561 3723 3946 4156 4324 5235 5274 5905 6649 6909 7011 7193

## [1] "Womens Record Numbers by Year"

## 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012   
## 2356 2166 2972 3334 3542 3899 4333 5435 5690 6397 8323 8853 9030 9730

Finally our datasets are set up correctly! The 1999 data no longer reads as one string. We can now proceed with analyzing the data.



Over time, we see the mean age of male racers decreases. Note that despite the mean age decreasing, the oldest participants increase gradually. We notice in 2001 to 2003, that a few unusually small ages appear for men.

## [1] "Missing Age Values for Male Racers by Year"

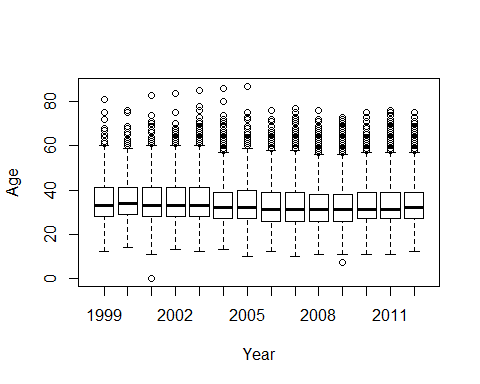
## 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012   
## 1 0 0 2 0 0 10 0 3 0 0 4 0 1

When we evaluate for missing values, we find a very small proportion of missing records exist. We’ll flag those that are missing with an index to potentially correct later. Next we assessed the run times for the men. All of the values look reasonable so we can be confident no outliers exist on the times for men.

## [1] "Men's run time summary statistics"

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 45.25 77.57 87.47 88.43 97.78 150.98

## Warning in FUN(X[[i]], ...): NAs introduced by coercion



We assessed the ages for women and notice a similar trend as the men had. Over time the mean age of female racers decreases. Unlike the men however, the oldest racers peaked around 2005 for the women then dropped siginificantly. There appears to be one potential age outlier for the female racers in 2001.

## [1] "Missing Age Values for female Racers by Year"

## 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012   
## 4 0 0 4 0 0 8 1 2 0 2 0 0 0

Similar to the men, a relatively small number of missing values exist for the women. Like the men, many of them exist in 2005. Next we’ll view all of the dubious ages for women in one code chunk. We see that most of the bad values are missing. There is one zero age record in 2001. An allegedly seven year old racer participated in 2009, but that is likely a legitimate record. Overall, the missing values are extremely low and won’t affect the remainder of the analysis, so we choose to ignore them.

Next we assessed the run times for the women. All of the values look reasonable so we can be confident no significant outliers exist on the times for women.

## [1] "Women's run time summary statistics"

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

Next we created a function to convert the times into a more usable format.

To make the datasets more functional, we created a function to convert the tables from the matrices into dataframes. Any values without run times will be dropped. After the function is created, it is run on the tables for both men and women. We notice one record from 2001 men’s results being dropped that had no run time.

## [1] "Dropping the following records in year 2001 for Men due to invalid times"  
## name home ag   
## "Peter HUI " "Silver Spring MD " "70 "   
## gun net time   
## "" "1:30:" NA

## Warning in data.frame(year = rep(year, nrow(Res)), sex = rep(sex,  
## nrow(Res)), : NAs introduced by coercion

Let’s check how many variables are missing. After converting to a dataframe, we notice that the missing ages are identical to the matrix tables. This gives us confidence that the conversion to dataframes was successful.

## [1] "men's missing ages"

## 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012   
## 1 0 0 2 0 0 10 0 3 0 0 4 0 1

## [1] "women's missing ages"

## 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012   
## 4 0 0 4 0 0 8 1 2 0 2 0 0 0

Next we ran a check to see how many hometown values are missing. We show no records for men or women. This is a good sign, but it’s possible the hometowns are stored as some type of text string even when missing and don’t convert into NAs. We’ll keep this in mind for later in the analysis.

## [1] "men's missing hometowns"

## 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012   
## 0 0 0 0 0 0 0 0 0 0 0 0 0 0

## [1] "women's missing hometowns"

## 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012   
## 0 0 0 0 0 0 0 0 0 0 0 0 0 0

When checking for missing run times, we notice only one potentially missing value that the function to convert to dataframe may have missed.

## [1] "men's missing run times"

## 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012   
## 0 0 1 0 0 0 0 0 0 0 0 0 0 0

## [1] "women's missing run times"

## 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012   
## 0 0 0 0 0 0 0 0 0 0 0 0 0 0

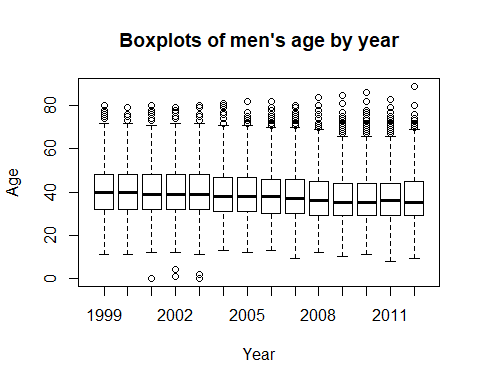
Next we store our dataframes for men and women into cbMen and cbWomen respectively.

## [1] 70070 6

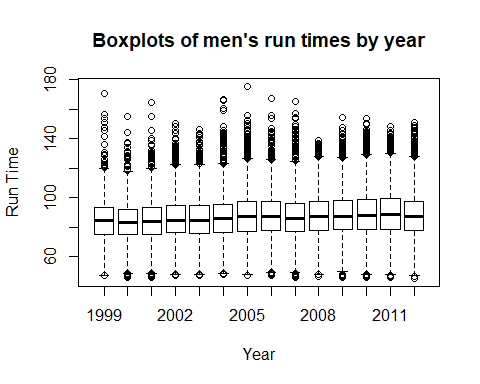
## [1] 75972 6

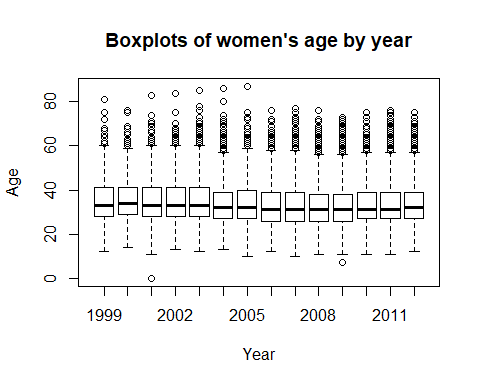
By calling head(menDF), we can take a look at our dataframes. We notice that all of the tables have uniform headers and the run times have converted correctly.

We again run some boxplots for the men’s age on the dataframes. The results look identical to the matrices.



We assess some boxplots for the men’s run times by year. We observe that the times increased gradually until around 2005, after which the times leveled off.



We look at the women’s ages by year in the dataframes and again notice it’s identical to the tables in the matrices. This gives us further validation that the data hasn’t undergone any significant changes during conversions. 

Here we perform a deeper dive into the men’s run times. All years show reasonable run times and only one missing value is observed. Run times are observed to be slower every year.

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 46.98 74.80 84.28 84.33 93.08 170.83

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 46.10 74.77 83.22 83.61 92.15 155.18

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 46.18 75.33 84.16 84.52 93.15 164.60 1

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 47.20 76.09 84.72 85.68 94.70 150.15

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 46.92 76.00 84.78 85.84 94.73 146.28

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 48.20 76.57 85.88 86.44 95.22 166.73

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 46.90 77.42 87.27 88.02 97.20 175.60

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 47.40 78.09 87.25 88.17 97.29 167.08

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 46.02 77.42 86.08 87.23 96.42 165.60

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 46.23 77.68 87.48 88.26 97.70 138.98

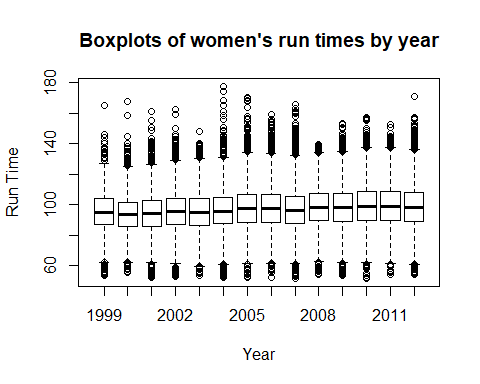
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 45.93 78.15 87.55 88.56 97.80 154.37

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 45.72 78.47 87.85 89.12 98.75 154.12

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 45.60 78.65 88.47 89.63 99.32 148.70

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 45.25 77.57 87.47 88.43 97.78 150.98

We ran the women’s run times by year and noticed a similar trend to the men in that they increase then level off after 2005.



Here we perform a deeper dive into the women’s run times. All years show reasonable run times and no missing values are observed. Like the men, run times are observed to be slower every year.

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 53.62 87.18 95.16 95.50 103.85 165.30

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 55.83 85.74 93.42 93.91 101.43 167.72

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 52.25 86.00 94.20 94.65 102.49 161.02

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 52.83 86.82 95.45 95.94 103.88 162.37

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 53.70 86.25 95.08 95.55 103.89 147.73

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 52.10 87.41 95.68 96.78 104.96 177.52

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 52.02 88.67 97.38 98.59 107.00 170.47

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 52.18 88.66 97.53 98.16 106.72 159.00

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 51.73 87.82 96.46 97.17 105.55 165.57

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 54.35 89.60 98.38 98.77 107.40 139.23

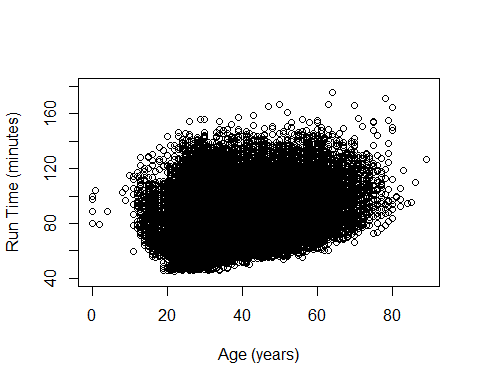
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 53.53 89.30 98.25 98.81 107.48 153.47

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 51.85 89.77 98.85 99.72 108.82 157.05

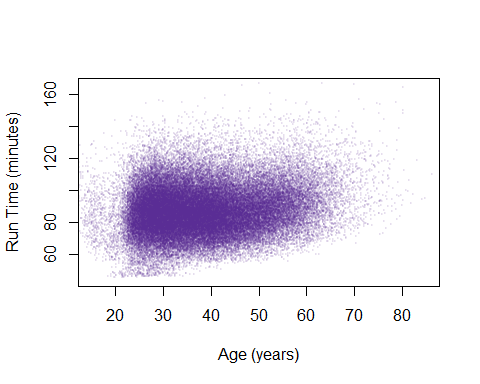
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 54.03 89.97 99.05 99.77 108.95 152.27

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

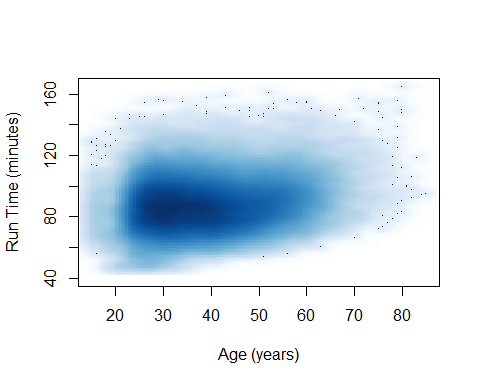
Next we produce a scatterplot of the run times by age for the men.



There’s a clear trend of longer run time as ages increase. However, the plot is a bit crowded so we produce one with smaller dots and a jitter affect to make it more readable.



This chart gives a clearer picture of the trend. There are some large clusters between around 70 to 100 minutes. We can create a density plot to better see where the concentration of times are.



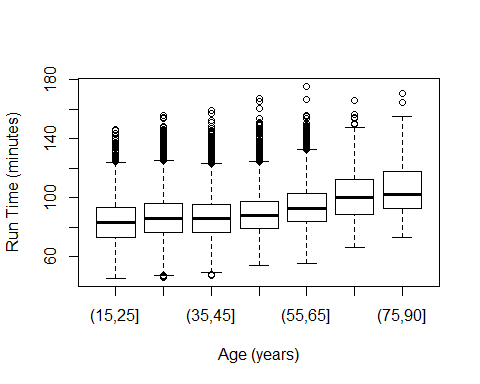
To get another view of how run times increase with age, we place the ages into buckets and create boxplots of run times.

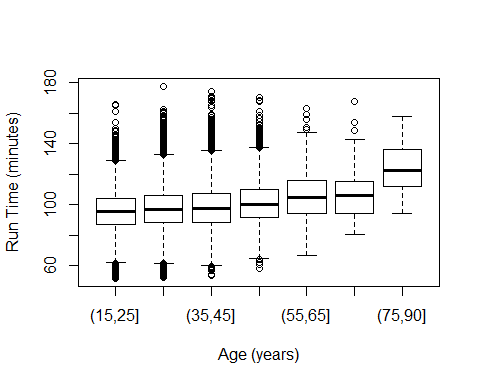
##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:gdata':  
##   
## combine, first, last

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union





We then boxplot the binned age groups. In the older buckets, the number of runners above the top line decreases. This indicates that older runners deviate to a shared mean and that once over 65, only serious runners participate.

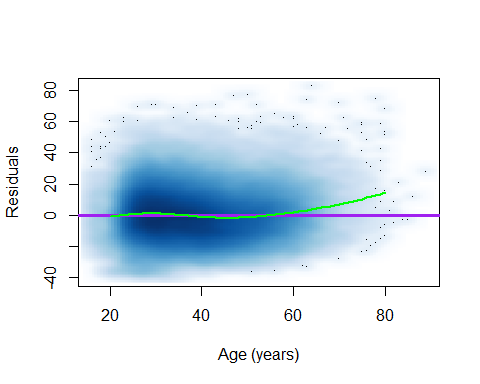
Next we preform a simple regression. We observe a coefficent of 0.225 for age. This confirms our earlier findings that as age increases, run times increase.

## (Intercept) age   
## 78.7567186 0.2252921

##   
## Call:  
## lm(formula = runTime ~ age, data = cbMenSub)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -40.333 -10.220 -0.952 9.102 82.425   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 78.75672 0.20770 379.18 <2e-16 \*\*\*  
## age 0.22529 0.00517 43.58 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 14.77 on 69804 degrees of freedom  
## Multiple R-squared: 0.02649, Adjusted R-squared: 0.02647   
## F-statistic: 1899 on 1 and 69804 DF, p-value: < 2.2e-16

## [1] "lm"

We used the aforementioned regression to create the green line below of predicted run times by age. We plot it along with the density scatter plot and notice that predicted run times remain around a zero residual, until age 60. After age 60, it shows seperation from the actual values and overpredicts run time.



Since the density is high from 80 to 100 minutes, a LOESS regression is run. LOESS creates a polynomial fit to the points, where more weight is given to points near the point whose response is being estimated and less weight to points further away. Furthermore, since seperation occurs around age 50, we create an additional parameter that measures the number of years older than 50 the runners are. Not that runners under 50 are given a value of zero in this over 50 parameter.

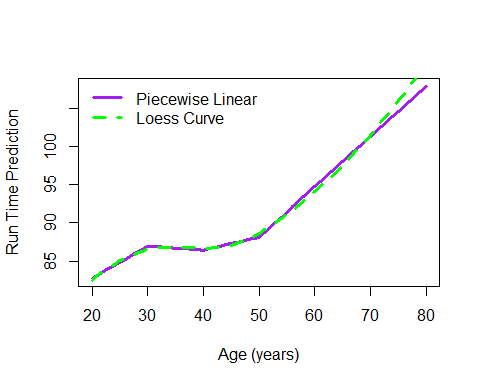
##   
## Call:  
## lm(formula = runTime ~ age + over50, data = cbMenSub)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -40.265 -10.098 -0.881 9.060 79.044   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 82.754891 0.265039 312.24 <2e-16 \*\*\*  
## age 0.105693 0.007147 14.79 <2e-16 \*\*\*  
## over50 0.563871 0.023371 24.13 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 14.71 on 69803 degrees of freedom  
## Multiple R-squared: 0.03454, Adjusted R-squared: 0.03451   
## F-statistic: 1248 on 2 and 69803 DF, p-value: < 2.2e-16

After performing the LOESS, we notice a large coefficient for runners over 50 and low coefficent for age relative to the simple regression. We want to take this a step further so we isolate to runners over age 30, 40, 50, and 60.

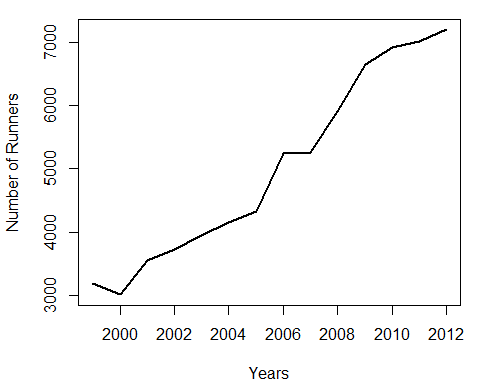
Once we have our data binned this way, we can perform a piecewise linear regression.

##   
## Call:  
## lm(formula = runTime ~ ., data = cbind(cbMenSub[, c("runTime",   
## "age")], overAge))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -40.921 -10.119 -0.885 9.023 78.965   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 74.228626 0.915248 81.102 < 2e-16 \*\*\*  
## age 0.424285 0.033207 12.777 < 2e-16 \*\*\*  
## over30 -0.477010 0.047778 -9.984 < 2e-16 \*\*\*  
## over40 0.221574 0.040666 5.449 5.09e-08 \*\*\*  
## over50 0.494432 0.052932 9.341 < 2e-16 \*\*\*  
## over60 -0.003601 0.077654 -0.046 0.963   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 14.7 on 69800 degrees of freedom  
## Multiple R-squared: 0.03593, Adjusted R-squared: 0.03586   
## F-statistic: 520.3 on 5 and 69800 DF, p-value: < 2.2e-16

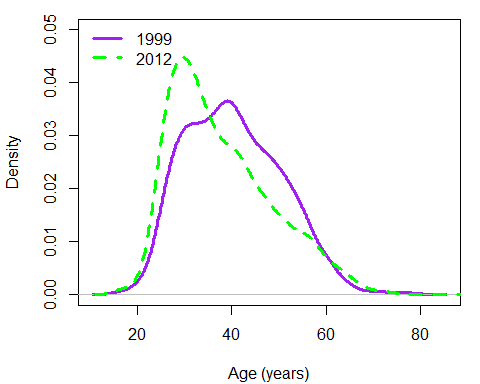
From the piecewise linear regression, we notice a large negative coefficeint for runners under age 30 of -0.477. Runners over age 40 and over age 50 have coefficients of 0.222 and 0.494 respectively. Runners over age 60 have a relatively flat coefficient. We proceed to plot predicted run time values using both the LOESS method and Piecewise linear regression.

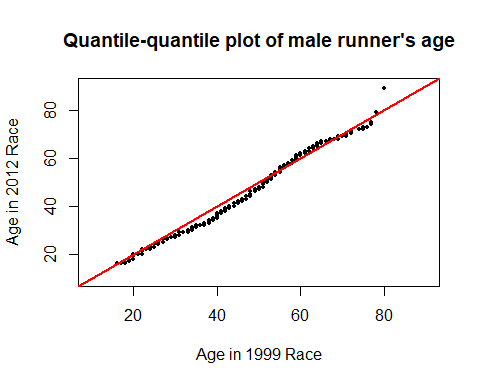


The next graph shows the number of runners each year. We notice a gradual increase from 3,000 to 7,000 runners. This along with the fact that ages were getting younger while run times were getting slower may be an indication that more amateur runners are signing up.

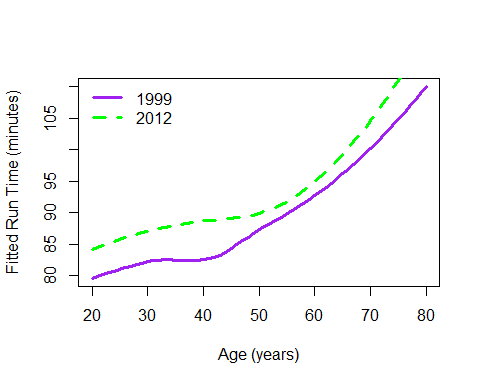


Next we plot age density by year for the men. We notice a shift to younger racers from 1999 to 2012. This gives us an interesting question to ask. Why does the average time increase although the average age decreases?



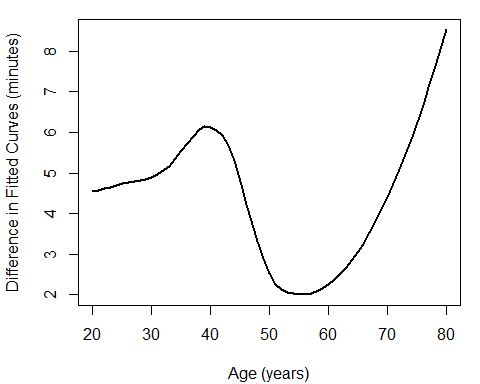


To answer the question of why the run times increase although age decreases, we need to assess the predicted run time by age in different years. We notice across the board the runners are slower. The gap is widest at age 40. Could this indicate that the runner’s include more amatuers? We can answer this by assessing what percentage of runners come from countries where professional runners are prevalent, such as Kenya.



When we aggregate the data, we observe that in 1999 twelve Kenyans ran. In 2012, only three Kenyans ran although the number of runners had more than doubled. This supports our theory that more amateurs and less professionals ran the race over the years.

Another visualization of how well the predictive models works is to look at the gap between predictions and actuals. We notice it’s easiest to predict times for runners between the ages of 50 and 60 and hardest for runners between age 70 and 80.



## Conclusion

In all practical scenarios, before any data analysis methods can be performed, the data needs to be first cleaned and formatted properly. This takes more time and effort than the actual analysis process. Real world data sets are almost never available to a data scientist in a perfect state and can present a myriad of challenges that must be overcome before starting the actual analysis. In this case study, we cleaned both men’s and women’s race results data files from the Cherry Blossom Ten Mile Run for the years 1999 through 2012. In our initial analysis of age distribution across all years from 1999 to 2012, for the general population of male race runners, we find a right-skewed distribution with the highest frequency of runners falling into the 30-40 years age bin. Using a stacked density plot to show a single curve for each year, there is a notable decrease in mean age from 1999 to recent years and earlier years tend to have a much more normal distribution in age compared to recent years.

Furthermore, our regression analyses confirmed what we learned from the literature review. While there is a steady peformance degradation as runners age, an inflection point occurs around age 50. This coincides with the research that shows consistent runners remained conditioned for endurance exercise into early old age. Consistency and good lifestyle choices can overcome age related peformance decrease.

## Future Work

As an extension of this effort, future works could possibly include modeling performance metrics and age distributions using different visualizations. We might consider evaluating age distribution across each year using residual plots or heat maps. Results would then be evaluated and compared against the density curves, quantile-quantile plots, and boxplots analyses to gain deeper insight and likely reinforce findings.

## References

1.) D. Lang and D. Nolan, *Data Science in R: A Case Studies Approach to Computation Reasoning and Problem Solving.* New York, New York: CRC Press. “Chapter 2: Modeling Runners’ Times in the Cherry Blossom Race”

2.) bthomasbailey, cherry-blossom-run (2017), GitHub repository, <https://github.com/bthomasbailey/cherry-blossom-run>

3.) Stefano Maria Iacus, Review of *Automated Data Collection with R – A Practical Guide to Web Scraping and Text Mining*, University of Milan <https://www.researchgate.net/profile/Stefano_Iacus/publication/284518787_Automated_Data_Collection_with_R_-_A_Practical_Guide_to_Web_Scraping_and_Text_Mining/links/56f1352d08ae5c367d4a9926/Automated-Data-Collection-with-R-A-Practical-Guide-to-Web-Scraping-and-Text-Mining.pdf>

4.) Leyk, Erley, Ridder, Leurs, Ruther, Wunderlich, Sievert, Baum, Essfeld, *Age-related Changes in Marathon and Half-Marathon Performances*, German Sports University of COlogne, 2 Central Institute of the Federal Armed Forces Medical Services Koblenz <https://www.researchgate.net/profile/Dieter_Leyk/publication/6236463_Age-related_Changes_in_Marathon_and_Half-Marathon_Performances/links/00b7d51a6137a2ad50000000.pdf>

5.) Mark Lebetkin, *Endurance Athletes Peak Later Than You Think* <https://www.theactivetimes.com/why-middle-age-isn-t-past-your-prime>