## **Introduction**

Beginning in 1973, the Cherry Blossom Ten-Mile Run is a popular training race that is held in Washington D.C. when the cherry blossoms are in bloom. It has become so popular that there were nearly 17,000 runners of all ages and levels in 2012 [1]. Each year, the results of the race are published online for anyone to access. For this analysis, we are interested in gaining insights into the runners’ times and ages at the Cherry Blossom Race using this publicly available dataset.

This analysis is an extension of the work completed in *Data Science in R: A Case Studies Approach to Computational Reasoning and Problem Solving*. The authors Deborah Nolan and Duncan Temple Lang scrape the aforementioned data into R and examine summary statistics of the male runners’ ages and race times. The main purpose of this analysis is to clean very unstructured data from the web and be able to use it for a simple statistical analysis [1].

For this case study, we will be detailing the Cherry Blossom Race data acquisition and cleaning process to answer Question 11 from the text. In brief, we will normalize the runners’ times for both male and female data, and visualize with density plots, quantile-quantile (Q-Q) plots and the statistics to compare the standard and normalized distributions of runners in 1999 and 2012. This will help us gain insight into the ages of individuals running this popular race 13 years apart. We will discover if any major differences exist and whether the passing of time has impacted the demographics of the clientele of the Cherry Blossom Ten-Mile Run.

## **Background**

The ten-mile Cherry Blossom Run has grown in popularity among locals and tourists due to the views of the popular Washington D.C. cherry blossom trees in full bloom during the time of the race. With a multitude of publicly available race data, we are able to gain insight into the runners of this race. The data available on the Cherry Blossom Race website provides runners’ places, names, home cities, ages, times, and paces for men and women beginning in 1999 [2].

The open source and publicly available nature of the Internet allows for anyone to gather enough data to run complex analyses. It is commonplace to find freely available data online with thousands or even millions of records. However, obtaining data from websites requires work to transform the data into a workable data set.

The data that we will use in this analysis is available on the Cherry Blossom Race’s website via a separate URL for each year’s results [2]. Therefore, it is necessary to perform web scraping to obtain the data of interest. Web scraping is the process of obtaining data from the Internet often through the website’s Hypertext Transfer Protocol (HTTP). This became popular as it is a much more efficient way to retrieve data from online sources than manually copying or downloading the data [3].

From a brief glance at the data that we will use for this report, it appears as though the quality and formatting of the data sets have evolved over the years. For this reason, extensive cleaning and formatting of the Cherry Blossom Race data will be necessary to perform a quality analysis. This process is detailed in the following section.

## **Method**

**Data acquisition**

To begin, we scraped the data from the Cherry Blossom Race’s website [2]. The results of each race are scraped from the individual URLs between years 1999 and 2012 for both men and women. We then saved the results from each year as a table in R. The lengths of each table, corresponding to the number of observations or number of runners in this case, can be seen in **Table 1**.

**Table 1: Number of runners each year**

|  |  |  |
| --- | --- | --- |
| **Year** | **Men** | **Women** |
| 1999 | 3194 | 2360 |
| 2000 | 3019 | 2169 |
| 2001 | 3628 | 2976 |
| 2002 | 3728 | 3339 |
| 2003 | 3952 | 3548 |
| 2004 | 4165 | 3907 |
| 2005 | 4336 | 4343 |
| 2006 | 5245 | 5445 |
| 2007 | 5284 | 5700 |
| 2008 | 5913 | 6406 |
| 2009 | 6654 | 8334 |
| 2010 | 6920 | 8864 |
| 2011 | 7020 | 9039 |
| 2012 | 7202 | 9739 |

**Table 1** reveals that the number of runners in the Cherry Blossom 10-mile race has increased every year in both men and women’s divisions since the start of the race in 1999, and more than doubled to 2012 (as shown in **Figure S1A** for male and **Figure S1B** for female runners).

Additionally, there have been more women runners than men runners every year since 2005. It appears as if the race has grown in popularity the most among women in recent years, with 9739 women running in 2012 compared to the 7202 men that ran the race in the same year.

Now that we have the data available, we can begin the cleaning process to ensure the data is in a workable format.

**Data cleaning**

The cleaning process began by identifying the header and the body of the tables of information. We then ensured that the information was placed into correct columns and extra white space was removed. The age values were formatted into numeric types and NA values were taken out. There were several rows of the data that was slightly off regarding spacing, so those observations were formatted to match the rest of the data. The time was then converted to minutes so that everything was in a standard format.

We can confirm that the data is cleaned and ready for analysis based on the age distributions. **Figure 1** displays the box plots of age by year for male and female runners before (**Figure 1A**) and after (**Figure 1B**) data cleaning. This shows relatively normally distributed data across all of the years, corrected for the year 2003 where all of the runners were originally listed as under ten years old and for the year 2006 where 25% of the runners were listed as being under ten years old (**Figure 1A**). Both of these errors were spacing errors in the data were corrected (**Figure 1B**).

|  |  |
| --- | --- |
| **Figure 1: Box plot of age by year to show a reasonable age distribution with data cleaning** | |
| **A: Before data cleaning** | **B: After data cleaning** |

**Exploring the run time for all male and female runners**

Now that we have completed extraction and cleaning of the data from the tables published on the Cherry Blossom Web site, we can begin to investigate the runners’ ages versus running times.

We will begin by examining the ages of male and female runners compared to their run times for all years. **Figure 2** displays the similar scatterplot results for men (**Figure 2A**) and women (**Figure 2B**). It is worth noting that in general the running times increase as age increases, as expected.

Most of the points appear as a dense blob in the scatter plot because so many points have been plotted on top of each other. The shape of the distribution is obscured because we cannot see which regions of the (age, run time) space are more densely populated.

We can also take notice of the vertical stripes in the plot. These are the result of runners’ ages being reported to the nearest year, which results in more over plotting. In the next section, we consider a few alterations to this default scatter plot that address the problem of over plotting.

|  |  |
| --- | --- |
| **Figure 2: Scatterplots for run time *vs*. age for male and female runners** | |
| **A: Male** | **B: Female** |
| There are several modifications we can make to the plot to ameliorate the effect of over plotting. We can reduce the size of the plotting symbol, use transparent colors for the plotting symbol, and add a small amount of random noise to the age variable. Alternatively, we can create a plot that reveals a smoothed version of the density of the points in each region. We can also make a series of boxplots instead of a scatter plot. Here, we demonstrate each of these approaches, as shown **in Figure 3**.  In **Figure 3A** and **3B**, the plots revise the simple scatter plots by changing the plotting symbol from a circle to a point, reducing the size of the plotting symbol, using a transparent color for the points, and adding a small amount of random noise to age. Now we see the shape of the high-density region containing most of the runners and the slight upward trend of time with increasing age.  This plot is much improved from the initial one in **Figure 2**. We can see where the bulk of the runners are, including what appears to be a slight upward curvature in run time as age increases and a skew distribution of run time given age. We can also see the small group of runners with very fast run times.  Consistent with scatter plots in **Figure 2**, the density plots shown in **Figure 3** reveals that most of the male (**Figure 3A**) and female (**Figure 3B**) runners are between the ages of about 25 to 55 and their times fall between 60 and 120 minutes in general.  Smoothed scatter plots of male (**Figure 3C**) and female (**Figure 3D**) runners offer an alternative to the scatter plots that use jittering and transparent color to ameliorate the over plotting. The shape of the high-density region has a very similar shape to the earlier plots. T  Additionally, the variations can also be seen from the smoothed scatter plots of the residuals from the fit of the simple linear model of run time to age for male (**Figure S2A**) and female (**Figure S2B**) runners who are 15 to 80 years old.  **Figure 3: Density plots of age vs. running time for men and women** | |
| **A: Male, circle to point** | **B: Female, circle to point** |
| **C: Male, smoothed** | **D: Female, smoothed** |

**Run time for subgroups of male and female runners with roughly the same age**

A very different approach to these scatter plots (**Figures 2** and **3**) is to graphically display summary statistics of run time for subgroups of runners with roughly the same age. Here in **Figure 4**, we group the runners into the sequences of boxplots showing the quartiles of time for men (**Figure 4A**) and women (**Figure 4B**) grouped into 10-year age intervals. As age increases, all the quartiles increase. However, the box becomes asymmetrical with age, which indicates that the upper quartile increases faster than the median and lower quartile. The distribution of run times between all of the age brackets is relatively normally distributed, but just like the men (**Figure 4A**), the running times increase overall as the age brackets increase. The female runners (**Figure 4B**) have more outliers at faster race times than the men, but these times are not quite as fast as the quickest men (**Figure 4A**). Younger runners generally have the faster race times between 60 and 140 minutes.

|  |  |
| --- | --- |
| **Figure 4: Boxplots of run time vs. age for male and female runners** | |
| **A: Male** | **B: Female** |

We can dive deeper into the age distributions of the runners by looking at the number of runners in each age group.

**Table 2** displays the number of runners in ages from 15-25, 25-35, 35-45, 45-55, 55-65, 65-75, and

|  |  |  |
| --- | --- | --- |
| **Table 2: Number of runners per age group** | | |
| **Age Group** | **Number of Male Runners** | **Number of Female Runners** |
| 15 – 25 | 5804 | 12,926 |
| 25 – 35 | 25,434 | 35,815 |
| 35 – 45 | 20,535 | 17,520 |
| 45 – 55 | 12,212 | 7515 |
| 55 – 65 | 5001 | 1849 |
| 65 – 75 | 752 | 183 |
| 75 – 90 | 69 | 15 |

75-90. The vast majority of male and female runners falls between the ages 25 – 55, and drastically decreases in number as the runners are below 25 and above 55. In addition, comparing the number of female racers to the number of male racers reveals that there were more females than males younger than 45, but more males from the age 45 and up.

**Piecewise linear models on the male and female runners’ ages vs. run time**

We can then run a linear model on the runners’ ages compared to their times to get a deeper understanding of the race.

The model created for the men’s ages versus running time can be represented by the linear equation . This means that for every one-year increase in age, the predicted running time increases by 0.225 minutes. This model is not a great fit for our data, with an adjusted r2 value of only 2.6%. We can attempt to find a better fitting model with a piecewise linear model.

We can get a better understanding of all age groups by expanding our linear model. We can represent this through the piecewise linear equation . This means that for every one-year increase in overall age, the predicted running time increases by 0.424 minutes and for every one-year increase in age over 30, the predicted running time decreases by 0.477. The estimated running time increases by 0.2215 minutes for every year increase over 40, increases by 0.495 minutes for every year increase over 50, and surprisingly decreases by 0.0047 for every year increase over 60. This model did improve slightly from the linear model with an adjusted r2 of 3.6%.

Comparatively, we also fit the women’s data using a piecewise linear model and a LOESS curve. The piecewise linear model is represented by the formula . This means that for every one-year increase in overall age, the predicted running time increases by 0.358 minutes and for every one-year increase in age over 30, the predicted running time decreases by 0.394 minutes. The estimated running time increases by 0.385 minutes for every year increase over 40, increases by 0.101 minutes for every year increase over 50, and decreases by 0.026 for every year increase over 60. The predicted increase in times are slightly less than the men’s, which may be evidence that women runners do not change as drastically in their race times as they get older.

**LOESS models on the male and female runners’ ages vs. run time**

In an attempt to find a more accurate model for this data, we can also run a local polynomial regression model (LOESS) on our data of men and women’s ages versus running times. This nonparametric model fits a smooth curve over our data, which may be a more accurate fit for this data since it does not assume linearity like the previous model [4]. **Figure 5** displays the fitted models (**Figure 5A** for male, and **Figure 5B** for female) for both the piecewise linear model (the solid purple line) and the LOESS curve (the dashed green line). The predictions for both models are very similar; the main difference is that the LOESS model is curved while the piecewise linear model is made up of straight lines.

Fitting a LOESS curve to the women’s data provides the model shown in **Figure 5B**. This displays the fitted models for both the piecewise linear model (the solid purple line) and the LOESS curve (the dashed green line). Again, the predictions for both models are very similar. The slope of the piecewise linear model is steeper after the age of 40 for the women compared to the men’s model, which doesn’t increase quite as fast until after age 50.

Another metric that we will use to compare the 1999 runners to the 2012 runners is the normalized running times, created by dividing the race time for each runner by the fastest race time of the same age. This allows us to compare the running times across all ages on the same scale. We can use these models to compare the running times of men in the year 1999 compared to 2012. The outcomes can be viewed in the Results section.

Again, the normalized values for women runners is then created by dividing each female runner’s time by the fastest time of their age. The piecewise linear model and the LOESS curve for both the standard race times and normalized race times can be used to compare the running times of women in the years 1999 and 2012. The results can be viewed in the following section.

|  |  |
| --- | --- |
| **Figure 5: Piecewise linear and LOESS curves fitted to run time vs. age for runners** | |
| **A: Male** | **B: Female** |

## **Results**

In this section, we will use LOESS curves, density plots, quantile-quantile plots, and summary statistics to compare the distribution of the standard and age-normalized times for runners in 1999 and 2012. We will show the results for both men and women.

**LOESS curves to performance for 1999 and 2012 runners**

**Figure 6** shows the fitted run time smoothed with the LOESS function by age for male (**Figure 6A**) and female (**Figure 6B**) runners in the 1999 and 2012 races for both standard race times (**Figures 6A, 6B**) and normalized (**Figures 6C, 6D**) race times.

In **Figure 6A** for male runners, the first insight that can be drawn from the standard time graph is that male runners in the 1999 race had a faster time for every age than runners in the 2012 race. This fact is rather surprising because it is a common belief that people now are faster on average than people from years ago. This is not being reflected in this race data. An insight that is not surprising is that as male runners’ ages increase, their times also increase. However, it is interesting to note that both lines are not straight lines. There is an increase then the slope decreases dramatically, very close to being zero or even negative, but then it increases again. After the near-zero slope period, the fitted time increases at a very high rate. This pattern occurs for both years but at different ages for each year.

In the 1999 race the near-zero slope starts at 33 and from age 33 to about 43 the fitted time is roughly the same value. After 43, the fitted run time skyrockets reaching its max at 110 minutes for an eighty-year-old male runner.

In the 2012 race, the line is closer to being straight and the near-zero slope period occurs at age 40. This slope is not zero as in the 1999 race, but it is not as drastic as the age groups before and after it. Again, at around age 50 the fitted time increases dramatically until it hits its max at age 80.

The different slopes and lines give evidence that there is a larger gap between middle age male runners now than in 1999. In 1999 male runners from age 33 to 43 all had a similar time. However, in the 2012 race while the times were closer than in other age groups there was a larger difference from age to age than in 1999.

In the normalized models (**Figure 6C** for male runners), a value of 1.0 would mean that the runners are predicted to be the fastest runners for their age. This graph reveals that the men in 1999 tended to be faster than those in 2012, which is consistent with the standard results discussed above. As age increases, the runners tend to run closer to the fastest runner’s time. This could be due to the fact that as age increases, fewer runners actually run the race. If there are only a few runners of each age, it is more likely that the predicted value will be close to 1.0.

These models are also supported by the difference in fitted curves, as shown in **Figures 6E**.

Similar to the men’s graph (**Figures 6A, 6C, and 6E**), the fitted run time was faster in the 1999 race than the 2012 women’s race for both standard race times and normalized race times, which can be seen in **Figures 6B, 6D and 6F**.

In **Figure 6B**, the female runner’s age also increased as their fitted time increased for the standard times. However, these graphs look very different than the men’s graphs. First the 1999 race line is almost a perfect straight line from age 35 to 80. Female runners’ times actually decreased from age 20 to 25 then remained constant until 35. This similar to the male 1999 race graph shows that between a certain age group there is no real distinction in time among the different ages in this group.

|  |  |
| --- | --- |
| **Figure 6: LOESS curves to performance for 1999 and 2012 runners with standard and normalized values** | |
| **A: Male, standard** | **B: Female, standard** |
| **C: Male, normalized** | **D: Female, normalized** |
| **E: Male, difference** | **F: Female, difference** |

However, this similar timed age group is much younger for females than males. In the 2012 we see a similar phenomenon where the fitted time increases than levels out than increases again. For this race it occurs starting at age 38 and ending at age 45. Just like the males, this near-zero slope age group is much older in 2012 than in the 1999 race. In the 2012 race the fitted time actual decreases slightly for woman in this age group.

|  |  |
| --- | --- |
| **Figure 7: Density curves for the age of male and female runners in 1999 and 2012 with standard and normalized values** | |
| **A: Male** | **B: Female** |
| **C: Male (Normalized)** | **D: Female (Normalized)** |

For the normalized fits (**Figure 6D**), the results are similar to the men’s results. As the women’s ages increase, the closer their running times get to the fastest runner of their same age. Overall, the runners in 1999 tended to be closer in time to the fastest runners of their ages than the runners in 2012.

These models are also supported by the difference in fitted curves, as shown in **Figures 6F**.

These results are also supported by the coefficients from longitudinal analysis of male (**Figure S5A**) and female (**Figure S5B**) athletes in scatter plots displaying the slope of the fitted line to each of the 300+ runners who competed in at least 8 Cherry Blossom road races. For example, nearly all of the coefficients for those male runners over 50 are positive. The typical size of this coefficient for a 50-year old male runner is about one minute per year.

We will dive deeper into the comparison of the men and women’s results in 1999 and 2012 with density plots, quantile-quantile plots (Q-Q plots), and summary statistics.

**Density plots for the age of runners in 1999 and 2012**

**Figure 7** displays the density plots of the male (**Figure 7A**) and female (**Female 7B**) runners’ ages in the two years. We see that while they are both slightly right skewed, the density plot for the year 2012 is a little higher and not quite as wide as that of 1999. This tells us that the race appealed to younger runners in 2012, while 1999 had a greater distribution of ages. This is consistent for the normalized running times, as shown in **Figures 7C** and **D**, although the distributions are slightly more normalized.

**Q-Q plots for the age of runners in 1999 and 2012**

Investigating the Q-Q plots (seen in **Figure 8**) of the male (**Figure 8A**) and female (**Female 8B**) runners’ ages and the normalized running times (**Figure 8C** for male, and **Figure 8D** for female) reveals that both of the ages in 1999 and 2012 and the times in 1999 and 2012 almost come from the same distribution. The running time distribution is slightly more off, as seen by the steeper slope of the residuals than the standard line in the plot.

**Figure 8** shows the Q-Q plots of the runners’ ages for both years (**Figure A**: Male; **Figure B**: Female; **Figure C**: Male with normalized data; **Figure D**: Female with normalized data). The distributions of the runners’ ages in both years are extremely similar, although they are both right skewed for both standard values and normalized values. This shows evidence the race appeals to a higher number of younger female runners in both years.

The Q-Q plots in **Figure 8** reveal that both the ages of runners in both years along with the normalized running times of both years come from the same distribution, as the residuals are relatively straight lines.

**Summary statistics for the male race times in 1999 and 2012**

Overall, the summary statistics shown in **Table 3** show consistent results with the LOESS fits shown in **Figure 6**. While the fastest runner in 2012 beat the fastest time from 1999, both the median and mean race times were slower in 2012 than in 1999. This result is consistent for both the standard race times and the normalized times. One explanation of this may be the increase in popularity of

|  |  |
| --- | --- |
|  | |
| **Figure 8: Q-Q plots for age and for normalized running times of runners in 1999 and 2012** | |
| **A: Male** | **B: Female** |
| **C: Male (Normalized)** | **D: Female (Normalized)** |

this race. At the conception of this race in 1999, a majority of the runners were competitive runners and often used this race as a training run for the Boston Marathon. Over time, the race became increasingly popular to the masses and many people in 2012 most likely ran the race for fun. It would be expected that the majority of the race times are slower when the runners are not competing for time.

**Summary statistics for the female race times in 1999 and 2012**

The summary statistics shown in **Table 4** show the overall results of the women’s races in 1999 and 2012. This is also consistent the results with the LOESS fits shown in **Figure 6**, where the

**Table 3: Summary statistics of the male race times in 1999 and 2012 (in minutes)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Statistic** | **1999 (Standard)** | **2012 (Standard)** | **1999 (Normalized)** | **2012 (Normalized)** |
| Minimum | 46.98 | 45.25 | 1.01 | 1.25 |
| 1st Quartile | 74.82 | 77.57 | 1.47 | 1.53 |
| Median | 84.29 | 87.47 | 1.49 | 1.62 |
| Mean | 84.35 | 88.44 | 1.51 | 1.62 |
| 3rd Quartile | 93.06 | 97.78 | 1.58 | 1.73 |
| Maximum | 170.83 | 150.98 | 1.64 | 1.75 |

women in 1999 tended to have faster race times than the women in 2012. All of the standard summary statistics for 1999 are faster times than that of 2012 and the normalized values are closer to 1.0 for the runners in 1999 than those in 2012. This again might be due to the higher number of non-competitive runners in the later years.

**Table 4: Summary statistics of the female race times in 1999 and 2012 (in minutes)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Statistic** | **1999 (Standard)** | **2012 (Standard)** | **1999 (Normalized)** | **2012 (Normalized)** |
| Minimum | 53.62 | 54.03 | 1.13 | 1.19 |
| 1st Quartile | 87.19 | 89.08 | 1.51 | 1.52 |
| Median | 95.17 | 98.03 | 1.58 | 1.62 |
| Mean | 95.53 | 99.02 | 1.55 | 1.61 |
| 3rd Quartile | 103.86 | 107.90 | 1.61 | 1.70 |
| Maximum | 165.30 | 170.97 | 1.63 | 1.73 |

## **Conclusions**

In this case study of modeling runners’ time in Cherry Blossom Race, we normalized the runners’ times for both male and female standardized and normalized data in 1999 and 2012, visualized with density plots, quantile-quantile (Q-Q) plots and then summarized in statistics.

The runners, mostly between ages 25 to 55, use more time as the ages increase. Interestingly, there are more female runner younger than age 45, but more male runners from the age 45 and up.

In the 1999 race both males and females had a faster fitted time than in the 2012 race. This gives evidence to the fact that in 1999 runners at every age had a smaller range of finish times than in 2012. Since the data was normalized by the fastest time for each age, values farther away from this minimum will result in a slower time than values closer to this minimum. There are more values farther away from the minimum in the 2012 race than in the 1999 race. Another way to say this is that the range of the fastest and slowest person was larger in 2012 than in 1999.

**LOESS curves to performance for 1999 and 2012**

For male runners, this LOESS fit of run time to age for year sits above the fit for 1999. The gap between these curves is about 5 minutes for most years. The exception is in the late 40’s to early 60’s where the curves are within 2–3 minutes of each other. Both curves have a similar shape. Comparatively for female runners, overall this LOESS fit of run time to age for year is above the fit for 1999. The gap between these curves is about 7 minutes maximum at age 35 and about 0 minutes at age 20. For late 40’s to early 60’s, the curves are also within 2–3 minutes of each other.

**Density and QQ-plots for the runner’s age distribution in 1999 and 2012**

For male runners, the density plots indicate the race appealed to younger runners in 2012, while 1999 had a greater distribution of ages for runners. Comparatively, the density plots indicate the runners had a quite similar age distribution in 1999 and 2012. These results were further confirmed with QQ-plots results with standardized and normalized data.

**Summary statistics for the race times in 1999 and 2012**

The summary statistics for the race times in 1999 and 2012 indicate the races have become increasingly popular and less competitive for male and female runners in Cherry Blossom Ten-Mile Run.

## **Future Work**

**Model expansion**

For both males and female runner in 2012 had a slower fitted time than the 1999 runners for every age. However, this does not mean they had a slower time overall. In the process used to create the normalized graphs, each runner’s time was normalized by the fastest time of a runner of the same age. An outlier would have a massive impact on this normalization process. If in the 2012 race there was one runner at every age that ran a much faster time than his or her peers than this would cause the normalized values to be larger than if all runners ran at a similar time. It would be interesting to compare the results of this analysis to the results of a different normalization process. We could normalize the data based on the slowest runner in each age group, the fastest runner overall, or the slowest runner overall. These analyses could also answer the questions: as a whole are people faster in 2012 than 1999? And are the fastest people in 2012 faster compared to average than the fastest compared to average in 1999?

**Understand each runner’s performance**

We will investigate further on each runner’s performance after multiple races changing with ages, as shown in **Figures S3** and **S4**. Our preliminary studies show very interesting patterns. For example, many middle-aged runners show a sharp increase in run time with age but that is not the case for all. Some of them improve and others change more slowly. Individual runner’s times series will help address this in future studies.

**Ethical considerations**

Since the data used in this analysis was scraped from the web, it is necessary to talk about the ethical implications of this process. Web scraping is a common technique that is widely used in the data science community today. However, the individuals that created the data (in this case the runners) are not aware that their data is being used for analysis. In this case, the runners’ information is fully available online. This includes very personal information such as their full name, city, and age. Most likely, the runners did not give explicit permission for this data to be available online; runners would probably have to opt-out in order to prevent their information to be available in the data sets. Our team definitely did not receive consent from the runners to use their data in this analysis. While this is technically legal as of now, there are major strides worldwide to protect individuals and their data.

The General Data Protection Regulation (GDPR) is a recently adopted law in the European Union (EU). This outlines strict guidelines for companies that wish to use individual’s data in an effort to protect citizens’ privacy and safety [5]. Under these laws, companies have to receive written consent in order to use an individual’s data and must be clear about what they are using the data for. Once the aforementioned purpose of the data is completed, the data must be destroyed. If any of the runners of the Cherry Blossom Race are EU citizens, we would not be able to use their data for this analysis since we did not receive consent to use their data for our analysis.

## While it is technically legal to use this data in the United States, it is worth considering the ethical and legal implications of web scraping for this project. Runners may not want their data to be used in analyses such as this one and the laws around data may change in the U.S. If the U.S. implements laws similar to the GDPR in the near future, we will have to obtain data from other sources.**References**

1. Nolan, D. and Temple Lang, D. (2015). Data Science in R: A Case Studies Approach to Computational Reasoning and Problem Solving. Boca Raton, FL: CRC Press (Chapter 2).
2. THE RACES. (n.d.). Retrieved from <http://www.cherryblossom.org/aboutus/results_list.php>
3. Web Scraping: How It All Started And Will Be. (n.d.). Retrieved January 29, 2019, from https://www.octoparse.com/blog/web-scraping-introduction
4. LOESS Curve Fitting (Local Polynomial Regression). (n.d.). Retrieved February 2, 2019, from https://www.statsdirect.com/help/nonparametric\_methods/loess.htm
5. Art. 1 GDPR – Subject-matter and objectives. (n.d.). Retrieved February 2, 2019, from https://gdpr-info.eu/art-1-gdpr/

## **Appendix - R code**

### -----------------------------------------------

## Section 0: Load the packages for this case study

### -----------------------------------------------

Packages <- c("xml2", "rvest", "stringr", "purrr", 'RColorBrewer', 'ggplot2', 'tidyr', 'broom', 'dplyr', 'ggthemes')

lapply(Packages, library, character.only = TRUE)

## ----------------------------------------------------------------------

## Section 1: Scraping the race results from the web (www.cherryblossom.org)

## ----------------------------------------------------------------------

## Use the 'read\_html() in the 'rvest' package to scrape

## The online table was previously inserted into a '<pre>' node from the HTML source

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

url <- paste0(ubase, 'results/1999/award.html')

doc <- read\_html(url)

pre\_node <- html\_nodes(doc, 'pre') # access all '<pre>' nodes in the document with the CSS selector 'pre'

txt <- html\_text(pre\_node) # extract the text content from this node

nchar(txt) # examine the contents of of 'txt'

str\_sub(txt, 1, 50)

str\_sub(txt, nchar(txt) - 50, nchar(txt))

els <- str\_split(txt, '\\r\\n')[[1]] # use '\r\n' to split up the 690904 characters into separate strings

length(els)

head(els, 3)

tail(els, 3)

## Formalize these codes into a function to take as input the URL for the Web page

## and return a character vector with one element per line,

## including the header lines and the rows in the table of results

extract\_res\_table <- function(url) {

read\_html(url) %>%

html\_nodes('pre') %>%

html\_text() %>%

str\_split('\\r\\n') %>%

.[[1]]

}

## Try this function with the 1999 men's results

m1999 <- extract\_res\_table(url)

identical(m1999, els)

## Construct the 'urls' vector from proper web addresses for all the years

men\_urls <- 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'

)

men\_urls <- paste0(ubase, men\_urls)

head(men\_urls, 3)

# Fix some issues ('\n' for year 1999, 'html\_nodes(font)')

extract\_res\_table <- function(url, year = 2001) {

# handle weird cases

if (year == 1999) {

read\_html(url) %>%

html\_nodes('pre') %>%

html\_text() %>%

str\_split('\\n') %>%

.[[1]]

} else if (year == 2000) {

read\_html(url) %>%

html\_nodes('font') %>%

.[[4]] %>%

html\_text() %>%

str\_split('\\r\\n') %>%

.[[1]]

} else {

read\_html(url) %>%

html\_nodes('pre') %>%

html\_text() %>%

str\_split('\\r\\n') %>%

.[[1]]

}

}

## Use map2() to call 'extract\_res\_table()' to check and find year 2009 still has problem

years <- 1999:2012

men\_tables <- map2(men\_urls, years, extract\_res\_table)

names(men\_tables) <- years

map\_int(men\_tables, length)

## Final modified extract\_res\_table()

extract\_res\_table <- function(url, year = 2001) {

selector <- if (year == 2000) 'font' else 'pre'

regexp <- if (year == 1999) '\\n' else '\\r\\n'

result <- read\_html(url) %>%

html\_nodes(selector)

if (year == 2000) result <- result[[4]]

result <- result %>%

html\_text()

if (year == 2009) return(result)

result %>%

str\_split(regexp) %>%

.[[1]]

}

## Read results into R and check the results

## Now, we have the function working for the web pages of men's results

men\_tables <- map2(men\_urls, years, extract\_res\_table)

names(men\_tables) <- years

map\_int(men\_tables, length)

## Try this on the women's pages

women\_urls <- 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 <- paste0(ubase, women\_urls)

women\_tables <- map2(women\_urls, years, extract\_res\_table)

names(women\_tables) <- years

## Modify extract\_res\_table for 2009 (no need to change)

extract\_res\_table <- function(url, year = 2001, female = TRUE) {

selector <- if (year == 2000) 'font' else 'pre'

regexp <- if (year == 1999) '\\n' else '\\r\\n'

result <- read\_html(url) %>%

html\_nodes(selector)

if (year == 2000) result <- result[[4]]

result <- result %>%

html\_text()

if (year == 2009 && female == FALSE) return(result)

result %>%

str\_split(regexp) %>%

.[[1]]

}

## Check the finally extracted tables for men and women and for all the years form 1999 to 2012

men\_tables <- map2(men\_urls, years, extract\_res\_table, female = FALSE)

women\_tables <- map2(women\_urls, years, extract\_res\_table, female = TRUE)

names(men\_tables) <- years

names(women\_tables) <- years

map\_int(men\_tables, length)

map\_int(women\_tables, length)

## Save the data in the text files for further processing

dir.create('men')

dir.create('women')

walk2(men\_tables,

paste('men', paste(years, 'txt', sep = '.'), sep = '/'),

writeLines)

walk2(women\_tables,

paste('women', paste(years, 'txt', sep = '.'), sep = '/'),

writeLines)

## ---------------------------------------------

## Section 2: Reading race result tables into R

## ---------------------------------------------

## Try read.table() and find 'skip=8'is not working

## m2012 <- read.table('men/2012.txt', skip = 8)

## Use readLines() and check the first 10 rows of the 2012 men's table

els2012 <- readLines('men/2012.txt')

head(els2012, 10)

## Use readLines() and check the first 10 rows of the 1999 men's table

els1999 <- readLines('men/1999.txt')

head(els1999, 10)

## Use the 2012 men's results as test case for developing the code to read in all files

eq\_idx <- str\_which(els2012, '^===') # search through the character strings in els2012

# Extract the key row and the row above it in the table, discard earlier rows

spacer\_row <- els2012[eq\_idx]

header\_row <- els2012[eq\_idx - 1]

body <- tail(els2012, -eq\_idx)

# Extract runner's age by converting the column names to lower case

header\_row <- str\_to\_lower(header\_row)

# Search through 'header\_row' for 'ag'

age\_location <- str\_locate(header\_row, 'ag')

age\_location # the location of runner's age (starts at 49 and ends at 50 in each row of the table)

# Use above information to extract each runner's age using 'str\_sub()'

age <- str\_sub(body, start = age\_location[1,1], end = age\_location[1,2])

head(age)

summary(as.numeric(age)) # age range from 9 to 89, and 2 did not report ages

# Find the locations of all of the blanks in the line of ‘=’ characters

blank\_locs <- str\_locate\_all(spacer\_row, ' ')[[1]][ ,1]

blank\_locs # Blank spaces are found at the 6th, 18th, 25th, 48th, ...

# Augment `blank\_locs` with 0 so the first column starts one character after 0

search\_locs <- c(0, blank\_locs)

## Extract all the columns with 'str\_sub()'

values <- map(body, str\_sub,

start = head(search\_locs, -1) + 1,

end = tail(search\_locs, -1) - 1)

## Encapsulate the task of finding the starting and ending positions of the columns into a function

find\_col\_locs <- function(spacer\_row) {

space\_locs <- str\_locate\_all(spacer\_row, ' ')[[1]][ ,1]

row\_length <- nchar(spacer\_row)

if (!str\_detect(spacer\_row, ' $')) {

return(c(0, space\_locs, row\_length + 1))

} else return(c(0, space\_locs))

}

## Encapsulate into a function the code to extract the locations of the desired columns.

## This function need the desired columns names, the header row (contains the column names),

## and the locations of the blanks in the separator row

select\_cols <- function(col\_names, header\_row, search\_locs) {

select\_col <- function(name, header\_row, search\_locs) {

start\_pos <- str\_locate(header\_row, name)[1,1]

if (is.na(start\_pos)) return(c(NA, NA))

index <- sum(start\_pos >= search\_locs)

c(search\_locs[index] + 1, search\_locs[index + 1] - 1)

}

map(col\_names, select\_col,

header\_row = header\_row,

search\_locs = search\_locs) %>%

do.call('rbind', .)

}

## Example of using this function to find age variable columns

search\_locs <- find\_col\_locs(spacer\_row)

age\_loc <- select\_cols('ag', header\_row, search\_locs)

ages <- map\_chr(body, str\_sub,

start = age\_loc[ ,1],

end = age\_loc[ ,2])

summary(as.numeric(ages))

## Use only the first few characters to uniquely identify the desired columns

short\_col\_names = c("name", "home", "ag", "gun", "net", "time")

## Want the values for that variable to be 'NA' if a file does not have one of the desired variables

loc\_cols <- select\_cols(short\_col\_names, header\_row, search\_locs)

values <- map(body, str\_sub,

start = loc\_cols[ ,1],

end = loc\_cols[ ,2]) %>%

do.call('rbind', .)

## Check the type of the return value

class(values)

## The results form a matrix of character strings

## Here to show the first few rows of the matrix

colnames(values) <- short\_col\_names

head(values)

## The 2012 table has a column for time and not gun and net times so the gun and net values are NA.

## Here to check the last few lines

tail(values)

## Wrap up the process of extracting the columns into a function so we can apply it to each year’s data

extract\_variables <- function(file, var\_names = c("name", "home", "ag", "gun", "net", "time")) {

# find the index of the row with =s

eq\_idx <- str\_which(file, '^===')

# extract the two key rows and the data

spacer\_row <- file[eq\_idx]

header\_row <- file[eq\_idx - 1] %>% str\_to\_lower()

body <- tail(file, - eq\_idx)

# get starting and ending positions of variables

search\_locs <- find\_col\_locs(spacer\_row)

loc\_cols <- select\_cols(var\_names, header\_row, search\_locs)

values <- map(body, str\_sub, loc\_cols) %>% do.call('rbind', .)

colnames(values) <- var\_names

invisible(values)

}

## Read the lines of the tables for men into R

mfilenames <- list.files('men', pattern = '.txt$', full.names = TRUE)

men\_files <- map(mfilenames, readLines)

names(men\_files) <- str\_match(mfilenames, 'men/(.\*).txt')[ ,2]

## Read the lines of the tables for women into R

wfilenames <- list.files('women', pattern = '.txt$', full.names = TRUE)

women\_files <- map(wfilenames, readLines)

names(women\_files) <- str\_match(wfilenames, 'women/(.\*).txt')[ ,2]

## Apply the 'extract\_variables()' function to 'men\_files 'to obtain a list of character matrices

men\_res\_mat <- map(men\_files, extract\_variables)

length(men\_res\_mat)

map\_int(men\_res\_mat, nrow)

## There is a problem with one of the women's files

## In 2001, the separator row of '=' characters does not exist

men\_file\_2001 <- men\_files$'2001'

women\_file\_2001 <- women\_files$'2001'

eq\_idx\_2001 <- str\_which(men\_file\_2001, '^===')

spacer\_row\_2001 <- men\_file\_2001[eq\_idx\_2001]

header\_row\_2001 <- men\_file\_2001[eq\_idx\_2001 - 1] %>% str\_to\_lower()

women\_files$'2001'[2] <- header\_row\_2001

women\_files$'2001'[3] <- spacer\_row\_2001

## Get reasonable values for the number of rows in our matrices

women\_res\_mat <- map(women\_files, extract\_variables)

length(women\_res\_mat)

map\_int(women\_res\_mat, nrow)

## ------------------------------------------------------------------

## Section 4: Clean data and reformat variables to be different types

## ------------------------------------------------------------------

mfilenames = paste("men/", 1999:2012, ".txt", sep = "")

menFiles = lapply(mfilenames, readLines)

names(menFiles) = 1999:2012

wfilenames = paste("women/", 1999:2012, ".txt", sep = "")

womenFiles = lapply(wfilenames, readLines)

names(womenFiles) = 1999:2012

## Convert the list of character matrices into an appropriate format for analysis

men\_res\_mat <- map(men\_files, extract\_variables)

length(men\_res\_mat)

map\_int(men\_res\_mat, nrow)

## Create the numeric variable 'age' with 'as.numeric()' ( example for the 2012 males)

age <- as.numeric(men\_res\_mat[['2012']][ ,'ag'])

## Check a few `age` values with tail()

tail(age)

## Check more thoroughly to confirm data extraction works as expected by summarizing each year’s ages

## Receive warning messages that our conversion of the character values for age into numeric resulted in `NA` values,

## meaning that some of the values do not correspond to numbers

age <- map(men\_res\_mat, ~ as.numeric(.x[ ,'ag']))

## Create side-by-side boxplots of the yearly distribution of the age of the runners

## Quick check on the reasonableness of the values

boxplot(age, ylab = "Age", xlab = "Year")

dev.copy(png,filename="./cs2Figures/age\_runners\_01.png", width=400, height=400);

dev.off ();

## There are problems for 2 years.

## All of the runners in 2003 were under 10

## More than 1 in 4 runners in 2006 were under 10

## In 2003, the age values are shifted to the right one space in comparison to the location of the ‘=’ characters

## This means that we are picking up only the digit in the tens place

head(men\_files$'2003')

## In 2006, some but not all of the rows have values that are off by one character

men\_files$'2006'[2200:2205]

## Modify the line in `select\_cols()` that locates the end of a column to include the blank position

select\_cols <- function(col\_names, header\_row, search\_locs) {

select\_col <- function(name, header\_row, search\_locs) {

start\_pos <- str\_locate(header\_row, name)[1,1]

if (is.na(start\_pos)) return(c(NA, NA))

index <- sum(start\_pos >= search\_locs)

c(search\_locs[index] + 1, search\_locs[index + 1])

}

map(col\_names, select\_col,

header\_row = header\_row,

search\_locs = search\_locs) %>%

do.call('rbind', .)

}

## Remove blanks with rgular expressions

men\_res\_mat <- map(men\_files, extract\_variables)

women\_res\_mat <- map(women\_files, extract\_variables)

age <- map(men\_res\_mat, ~ as.numeric(.x[ ,'ag']))

## Boxplots show reasonable age distributions and the problem above has cleaned up

boxplot(age, ylab = "Age", xlab = "Year")

dev.copy(png,filename="./cs2Figures/age\_runners\_02.png", width=400, height=400);

dev.off ();

## For the warning messages "NAs introduced by coercion", count the number of 'NA' values in each year

map\_int(age, ~ sum(is.na(.)))

## Check why year 2001 has 62 'NA's for age

age2001 <- age$'2001'

## Add an offset to the lication of the 'NA's in'age2001'

offset <- str\_which(men\_files$'2001', '^===')

offset

## Find the lines in the original file that have the bad age values

## All rows are empty except the footnote row

bad\_age\_idx <- which(is.na(age2001)) + offset

tail(men\_files$'2001'[bad\_age\_idx])

## Find where in the table are these rows located

bad\_age\_idx

## Modify the extraction by checking for blank rows and removing them

blanks <- str\_which(men\_files[['2001']], '^[[:blank:]]\*$')

## modified 'extract\_variables()'function

extract\_variables <- function(file, var\_names = c("name", "home", "ag", "gun", "net", "time")) {

# find the index of the row with =s

eq\_idx <- str\_which(file, '^===')

# extract the two key rows and the data

spacer\_row <- file[eq\_idx]

header\_row <- file[eq\_idx - 1] %>% str\_to\_lower()

# find blank lines

blanks <- str\_which(file, '^[[:blank:]]\*$')

# find comments

comments <- str\_which(file, '^[[:blank:]]\*[#\\\*]')

# remove header, blank lines, and comments

body <- file[-c(1:eq\_idx, blanks, comments)]

# get starting and ending positions of variables

search\_locs <- find\_col\_locs(spacer\_row)

loc\_cols <- select\_cols(var\_names, header\_row, search\_locs)

values <- map(body, str\_sub, loc\_cols) %>% do.call('rbind', .)

colnames(values) <- var\_names

invisible(values)

}

## After additional cleaning, the 61 'NA' are gone, but not others

men\_res\_mat <- map(men\_files, extract\_variables)

women\_res\_mat <- map(women\_files, extract\_variables)

map\_int(age, ~ sum(is.na(.)))

## Find which runners have an age under 5 and look at their records in the original table

which(age2001 < 5)

men\_files[['2001']][which(age2001 < 5) + offset]

## Create time variable()

char\_time <- men\_res\_mat[['2012']][ ,'time']

head(char\_time)

tail(char\_time)

## Use str\_split() to split each character string up into its parts

time\_pieces <- str\_split(char\_time, ':')

## Check heand() and ()

head(time\_pieces, 1)

tail(time\_pieces, 1)

## Convert these elements to numeric values

## and combine them into one value that reports time in minutes

run\_time <- map\_dbl(time\_pieces, function(x) {

x <- as.numeric(x)

if (length(x) == 2) x[1] + x[2] / 60

else 60 \* x[1] + x[2] + x[3] / 60

})

## Check with summary()

summary(run\_time)

## Encapsulate this conversion into a function called 'convert\_time()'

convert\_time <- function(t) {

time\_pieces <- str\_split(t, ':')

map\_dbl(time\_pieces, function(x) {

x <- as.numeric(x)

if (length(x) == 2) x[1] + x[2] / 60

else 60 \* x[1] + x[2] + x[3] / 60

})

}

## Wrap these conversions into a function creat\_df()

create\_df <- function(res, year, sex) {

# determine which time to use

use\_time <- if (!is.na(res[1,'net'])) {

res[ ,'net']

} else if (!is.na(res[1,'gun'])) {

res[ ,'gun']

} else {

res[ ,'time']

}

run\_time <- convert\_time(use\_time)

results <- data.frame(year = year,

sex = sex,

name = res[ ,'name'],

home = res[ ,'home'],

age = as.numeric(res[ ,'ag']),

time = run\_time)

invisible(results)

}

## Apply to the character matrices in menResMat and return a data frame with variables for analysis

men\_df <- map2(men\_res\_mat, 1999:2012, create\_df, sex = 'm')

## There are a large number of 'NA' in 2007, 2009, 2010 and all of the tume values for 2006

map\_dbl(men\_df, ~ sum(is.na(.x$time)))

## These are caused by runners who completed half the race but have no final times

## and by runners who have a footnote after their time

men\_files[['2007']][c(8, 5280)]

## Modify 'create\_df()'' to eliminate the footnote symbols

create\_df <- function(res, year, sex) {

use\_time <- if (!is.na(res[1,'net'])) {

res[ ,'net']

} else if (!is.na(res[1,'gun'])) {

res[ ,'gun']

} else {

res[ ,'time']

}

# remove #, \*, and blanks from time

use\_time <- str\_replace\_all(use\_time, '[#\\\*[:blank:]]', '')

# drop rows with no time

res <- res[use\_time != '', ]

run\_time <- convert\_time(use\_time[use\_time != ''])

results <- data.frame(year = year,

sex = sex,

name = res[ ,'name'],

home = res[ ,'home'],

age = as.numeric(res[ ,'ag']),

time = run\_time)

invisible(results)

}

## Apply this revised function to 'men\_res\_mat' to create data frame

men\_df <- map2(men\_res\_mat, 1999:2012, create\_df, sex = 'm')

## Most missing values are gone except for 2006

map\_dbl(men\_df, ~ sum(is.na(.x$time)))

## Look at the missing value for 2001

men\_df[['2001']] %>% filter(is.na(time))

i <- grep('Peter HUI', men\_files[['2001']])

men\_files[['2001']][(i-1):(i+1)]

## This runner's time got entered incorrectly

## Fill in that value manually

men\_df[['2001']]$time[2250] <- 91 + 47/60

## As for 2006, the problem can be seen when we look at the header lines of that file

men\_files[['2006']][7:11]

## Alter the separator line directly and rerun 'extract\_variables()'

spacer\_row\_2006 <- men\_files[['2006']][8]

str\_sub(spacer\_row\_2006, 64, 64) <- ' '

men\_files[['2006']][8] <- spacer\_row\_2006

men\_files[['2006']][7:11]

# library(dplyr)

## combine the race results for all years and men into one data frame

men\_res\_mat <- map(men\_files, extract\_variables)

men\_df <- map2\_dfr(men\_res\_mat, 1999:2012, create\_df, sex = 'm') %>%

mutate\_if(is\_character, funs(str\_trim(., side = 'both'))) %>%

mutate(name = str\_to\_title(name))

men\_df[which(men\_df$year == 2001 & men\_df$name == 'Peter Hui'),'time'] <- 91 + 47/60

## Write men\_df.scv file for future processing

write.csv(men\_df, 'men/men\_df.csv')

## Check the dimension of our amalgamated data frame

dim(men\_df)

## Apply this revised function to 'women\_res\_mat' to create data frame

women\_df <- map2(women\_res\_mat, 1999:2012, create\_df, sex = 'f')

## There are a large number of 'NA' in 2007, 2009, 2010 and all of the tume values for 2006

map\_dbl(women\_df, ~ sum(is.na(.x$time)))

## As for 2006, the problem can be seen when we look at the header lines of that file

women\_files[['2006']][7:11]

## Alter the separator line directly and rerun 'extract\_variables()'

spacer\_row\_2006 <- women\_files[['2006']][8]

str\_sub(spacer\_row\_2006, 64, 64) <- ' '

women\_files[['2006']][8] <- spacer\_row\_2006

women\_files[['2006']][7:11]

## Clean and reformat women's race results and save to women\_df.csv

women\_res\_mat <- map(women\_files, extract\_variables)

women\_files[['2006']][8] <- spacer\_row\_2006

women\_df <- map2\_dfr(women\_res\_mat, 1999:2012, create\_df, sex = 'f')

women\_df[grep('^Marie-Laure Poir', women\_df$name),'name'] <- 'Marie-Laure Poire'

women\_df <- women\_df %>%

mutate\_if(is\_character, funs(str\_trim(., side = 'both'))) %>%

mutate(name = str\_to\_title(name))

## Write women\_df.scv file for future processing

write.csv(women\_df, 'women/women\_df.csv')

## Check the dimension of our amalgamated data frame

dim(women\_df)

plot(time ~ age, data = men\_df,

ylim = c(40, 180), xlab = "Men ages (years)", ylab = "Run Time (minutes)",

col = "#3366FF")

dev.copy(png,filename="./cs2Figures/scatter\_men.png", width=400, height=400);

dev.off ();

##

plot(time ~ age, data = women\_df,

ylim = c(40, 180), xlab = "Women ages (years)", ylab = "Run Time (minutes)",

col = "#CC6666")

dev.copy(png,filename="./cs2Figures/scatter\_women.png", width=400, height=400);

dev.off ();

# display.brewer.all()

# Purples8 = brewer.pal(9, "Purples")[8]

# Purples8

# Purples8A = paste(Purples8, "14", sep = "")

men\_df %>%

filter(age > 5) %>%

ggplot(aes(age, time)) +

geom\_jitter(shape = '.', size = 2, alpha = 0.2, height = 0, width = 0.5, color = '#54278f')

dev.copy(png,filename="./cs2Figures/scatter5\_men.png", width=400, height=400);

dev.off ();

women\_df %>%

filter(age > 5) %>%

ggplot(aes(age, time)) +

geom\_jitter(shape = '.', size = 2, alpha = 0.2, height = 0, width = 0.5, color = '#54278f')

dev.copy(png,filename="./cs2Figures/scatter5\_women.png", width=400, height=400);

dev.off ();

men\_df %>%

filter(age > 5) %>%

ggplot(aes(age, time)) +

stat\_density\_2d(aes(fill = ..density..), geom = 'raster', contour = FALSE) +

scale\_fill\_gradientn(colors = c('white', 'dodgerblue3', 'dodgerblue4'), values = c(0, 0.5, 1)) +

geom\_hline(yintercept = 0, color = 'navy')

dev.copy(png,filename="./cs2Figures/scatter5\_density\_men.png", width=400, height=400);

dev.off ();

women\_df %>%

filter(age > 5) %>%

ggplot(aes(age, time)) +

stat\_density\_2d(aes(fill = ..density..), geom = 'raster', contour = FALSE) +

scale\_fill\_gradientn(colors = c('white', 'dodgerblue3', 'dodgerblue4'), values = c(0, 0.5, 1)) +

geom\_hline(yintercept = 0, color = 'navy')

dev.copy(png,filename="./cs2Figures/scatter5\_density\_women.png", width=400, height=400);

dev.off ();

## Section 5: Exploring the run time for all runners (men and women)

# get normalized value

men\_df\_99 <- men\_df %>%

filter(!is.na(age), age > 19, age < 81, year=='1999')

men\_df\_12 <- men\_df %>%

filter(!is.na(age), age > 19, age < 81, year=='2012')

women\_df\_99 <- women\_df %>%

filter(!is.na(age), age > 19, age < 81, year=='1999')

women\_df\_12 <- women\_df %>%

filter(!is.na(age), age > 19, age < 81, year=='2012')

# find fastest runner each year

getmin<-function(time){

time/min(time)

}

m\_fastest\_99 <- tapply(men\_df\_99$time, men\_df\_99$age, getmin)

m\_fastest\_12 <- tapply(men\_df\_12$time, men\_df\_12$age, getmin)

w\_fastest\_99 <- tapply(women\_df\_99$time, women\_df\_99$age, getmin)

w\_fastest\_12 <- tapply(women\_df\_12$time, women\_df\_12$age, getmin)

# men normalized values

toappend <- list()

i <- 1

for (j in m\_fastest\_99) {

age <- names(m\_fastest\_99[i])

val <- replicate(length(j), age)

toappend <- append(toappend, val)

i = i+1

}

agedf <- as.data.frame(matrix(unlist(toappend)))

timedf <- as.data.frame(matrix(unlist(m\_fastest\_99)))

men\_99\_norm <- cbind(agedf, timedf)

names(men\_99\_norm) <- c("Age", "Time")

toappend <- list()

i <- 1

for (j in m\_fastest\_12) {

age <- names(m\_fastest\_12[i])

val <- replicate(length(j), age)

toappend <- append(toappend, val)

i = i+1

}

agedf <- as.data.frame(matrix(unlist(toappend)))

timedf <- as.data.frame(matrix(unlist(m\_fastest\_12)))

men\_12\_norm <- cbind(agedf, timedf)

names(men\_12\_norm) <- c("Age", "Time")

# women normalized values

toappend <- list()

i <- 1

for (j in w\_fastest\_99) {

age <- names(w\_fastest\_99[i])

val <- replicate(length(j), age)

toappend <- append(toappend, val)

i = i+1

}

agedf <- as.data.frame(matrix(unlist(toappend)))

timedf <- as.data.frame(matrix(unlist(w\_fastest\_99)))

women\_99\_norm <- cbind(agedf, timedf)

names(women\_99\_norm) <- c("Age", "Time")

toappend <- list()

i <- 1

for (j in w\_fastest\_12) {

age <- names(w\_fastest\_12[i])

val <- replicate(length(j), age)

toappend <- append(toappend, val)

i = i+1

}

agedf <- as.data.frame(matrix(unlist(toappend)))

timedf <- as.data.frame(matrix(unlist(w\_fastest\_12)))

women\_12\_norm <- cbind(agedf, timedf)

names(women\_12\_norm) <- c("Age", "Time")

# Get age categories

men\_df\_agecat <- men\_df %>%

filter(time > 30, !is.na(age), age > 15) %>%

mutate(age\_cat = cut(age, breaks = c(seq(15, 75, 10), 90)))

table(men\_df\_agecat$age\_cat)

## Plot of the Number of Male Runners by Year.

## This plot shows that the number of male runners in the Cherry Blossom 10-mile race has more than doubled from 1999 to 2012

men\_df\_agecat %>%

ggplot(aes(factor(year))) +

geom\_bar(width = 0.6, fill='steelblue', color='steelblue') +

labs(x = 'Year', y = 'Number of Male Runners')

dev.copy(png,filename="./cs2Figures/barplot\_men.png", width=400, height=400);

dev.off ();

## The distribution of performance for men 1999 races

men\_df\_agecat %>%

filter(year == 1999) %>%

select(time) %>%

summary()

## The distribution of performance for men 2012 races

men\_df\_agecat %>%

filter(year == 2012) %>%

select(time) %>%

summary()

## Density curves for the age of male runners in 1999 and 2012

men\_df\_agecat %>%

filter(year %in% c(1999, 2012)) %>%

ggplot(aes(age, color = factor(year))) +

geom\_line(stat = 'density') +

scale\_color\_few(name = 'Year') +

labs(x = 'Age', y = 'Density')

dev.copy(png,filename="./cs2Figures/density\_men\_1999\_2012.png", width=400, height=400)

dev.off ();

# normalized

plot(density(men\_99\_norm$Time), col="purple", lwd = 3, main=" ")

lines(density(men\_12\_norm$Time), col="green", lwd = 3, lty = 3, )

legend("topright", col = c("green", "purple"), lty = 3:1, lwd = 3, legend = c("2012m", "1999m"))

dev.copy(png,filename="./cs2Figures/density\_men\_1999\_2012\_normalized.png", width=400, height=400)

dev.off ();

# normalized

plot(density(women\_99\_norm$Time), col="purple", lwd = 3, main=" ")

lines(density(women\_12\_norm$Time), col ="green", lwd = 3, lty = 3)

legend("topright", col = c("green", "purple"), lty = 3:1, lwd = 3, legend = c("2012w", "1999w"))

dev.copy(png,filename="./cs2Figures/density\_women\_1999\_2012\_normalized.png", width=400, height=400);

dev.off ();

## Compare these two distributions with a quantile-quantile plot

pts <- qqplot(men\_df\_agecat$age[men\_df\_agecat$year == 1999],

men\_df\_agecat$age[men\_df\_agecat$year == 2012],

plot.it = FALSE) %>% as\_data\_frame()

pts %>%

ggplot(aes(x, y)) +

geom\_point(fill='steelblue', color='steelblue') +

geom\_abline(slope = 1, intercept = 0) +

labs(x = 'Age in 1999', y = 'Age in 2012')

dev.copy(png,filename="./cs2Figures/qqplot\_men\_1999\_2012.png",width=400, height=400);

dev.off ();

men\_df\_agecat <- men\_df %>%

filter(time > 30, !is.na(age), age > 15) %>%

mutate(age\_cat = cut(age, breaks = c(seq(15, 75, 10), 90)))

table(men\_df\_agecat$age\_cat)

men\_df\_agecat %>%

ggplot(aes(age\_cat, time)) +

geom\_boxplot(fill='white', color='steelblue') +

labs(x = 'Age (years)', y = 'Run time (minutes)')

dev.copy(png,filename="./cs2Figures/boxplot\_men.png",width=400, height=400);

dev.off ();

# Normalized

pts <- qqplot(men\_99\_norm$Time,

men\_12\_norm$Time,

plot.it = FALSE) %>% as\_data\_frame()

pts %>%

ggplot(aes(x, y)) +

geom\_point(fill='steelblue', color='steelblue') +

geom\_abline(slope = 1, intercept = 0) +

labs(x = 'Time in 1999', y = 'Time in 2012')

dev.copy(png,filename="./cs2Figures/qqplot\_men\_1999\_2012\_normalized.png",width=400, height=400);

dev.off ();

women\_df\_agecat <- women\_df %>%

filter(time > 30, !is.na(age), age > 15) %>%

mutate(age\_cat = cut(age, breaks = c(seq(15, 75, 10), 90)))

table(women\_df\_agecat$age\_cat)

women\_df\_agecat %>%

ggplot(aes(age\_cat, time)) +

geom\_boxplot(fill='white', color='#CC6666') +

labs(x = 'Age (years)', y = 'Run time (minutes)')

dev.copy(png,filename="./cs2Figures/boxplot\_women.png",width=400, height=400);

dev.off ();

# Normalized

pts <- qqplot(women\_99\_norm$Time,

women\_12\_norm$Time,

plot.it = FALSE) %>% as\_data\_frame()

pts %>%

ggplot(aes(x, y)) +

geom\_point(fill='steelblue', color='#CC6666') +

geom\_abline(slope = 1, intercept = 0) +

labs(x = 'Time in 1999', y = 'Time in 2012')

dev.copy(png,filename="./cs2Figures/qqplot\_women\_1999\_2012\_normalized.png",width=400, height=400);

dev.off ();

women\_df\_agecat <- women\_df %>%

filter(time > 30, !is.na(age), age > 15) %>%

mutate(age\_cat = cut(age, breaks = c(seq(15, 75, 10), 90)))

table(women\_df\_agecat$age\_cat)

women\_df\_agecat %>%

ggplot(aes(age\_cat, time)) +

geom\_boxplot(fill='white', color='#CC6666') +

labs(x = 'Age (years)', y = 'Run time (minutes)')

dev.copy(png,filename="./cs2Figures/boxplot\_women\_normalized.png",width=400, height=400);

dev.off ();

## Line plot of the number of female runners by year.

## This plot shows that the number of female runners in the Cherry Blossom 10-mile race has more than 4-folds from 1999 to 2012

women\_df\_agecat %>%

ggplot(aes(factor(year))) +

geom\_bar(width = 0.6, fill='#CC6666', color='#CC6666') +

labs(x = 'Year', y = 'Number of Female Runners')

dev.copy(png,filename="./cs2Figures/barplot\_women.png", width=400, height=400);

dev.off ();

## The distribution of performance for women 1999 races

women\_df\_agecat %>%

filter(year == 1999) %>%

select(time) %>%

summary()

## The distribution of performance for women 2012 races

women\_df\_agecat %>%

filter(year == 2012) %>%

select(time) %>%

summary()

## Density curves for the age of female runners in 1999 and 2012

women\_df\_agecat %>%

filter(year %in% c(1999, 2012)) %>%

ggplot(aes(age, color = factor(year))) +

geom\_line(stat = 'density') +

scale\_color\_few(name = 'Year') +

labs(x = 'Age', y = 'Density')

dev.copy(png,filename="./cs2Figures/density\_women\_1999\_2012.png", width=400, height=400);

dev.off ();

## Compare these two distributions with a quantile-quantile plot

pts <- qqplot(women\_df\_agecat$age[women\_df\_agecat$year == 1999],

women\_df\_agecat$age[women\_df\_agecat$year == 2012],

plot.it = FALSE) %>% as\_data\_frame()

pts %>%

ggplot(aes(x, y)) +

geom\_point(fill='#CC6666', color='#CC6666') +

geom\_abline(slope = 1, intercept = 0) +

labs(x = 'Age in 1999', y = 'Age in 2012')

dev.copy(png,filename="./cs2Figures/qqplot\_women\_1999\_2012.png", width=400, height=400);

dev.off ();

## lm\_age for men

lm\_age\_m <- lm(time ~ age, data = men\_df\_agecat)

lm\_age\_m$coefficients

summary(lm\_age\_m)

## smoothScatter for men

smoothScatter(x = men\_df\_agecat$age, y = lm\_age\_m$residuals,xlab = "Age (years)", ylab = "Residuals")

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

m\_resid.lo = loess(resids ~ age,

data = data.frame(resids = residuals(lm\_age\_m),

age = men\_df\_agecat$age))

age20to80 = 20:80

m\_resid.lo.pr = predict(m\_resid.lo, newdata = data.frame(age = age20to80))

lines(x = age20to80, y = m\_resid.lo.pr, col = "green", lwd = 2)

dev.copy(png,filename="./cs2Figures/smoothScatter\_men.png", width=400, height=400);

dev.off ();

## lm\_age for women

lm\_age\_w <- lm(time ~ age, data = women\_df\_agecat)

lm\_age\_w$coefficients

summary(lm\_age\_w)

## smoothScatter for women

smoothScatter(x = women\_df\_agecat$age, y = lm\_age\_w$residuals,xlab = "Age (years)", ylab = "Residuals")

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

w\_resid.lo = loess(resids ~ age,

data = data.frame(resids = residuals(lm\_age\_w),

age = women\_df\_agecat$age))

age20to80 = 20:80

w\_resid.lo.pr = predict(w\_resid.lo, newdata = data.frame(age = age20to80))

lines(x = age20to80, y = w\_resid.lo.pr, col = "green", lwd = 2, lty = 3)

dev.copy(png,filename="./cs2Figures/smoothScatter\_women.png", width=400, height=400);

dev.off ();

men\_res\_lo <- loess(time ~ age, data = men\_df\_agecat)

women\_res\_lo <- loess(time ~ age, data = women\_df\_agecat)

age20to80 <- 20:80

men\_res\_lo\_pr <- predict(men\_res\_lo, data.frame(age = age20to80))

women\_res\_lo\_pr <- predict(women\_res\_lo, data.frame(age = age20to80))

over50m <- pmax(0, men\_df\_agecat$age - 50)

over50w <- pmax(0, women\_df\_agecat$age - 50)

lm\_over50m <- lm(time ~ age + over50m, data = men\_df\_agecat)

summary(lm\_over50m)

lm\_over50w <- lm(time ~ age + over50w, data = women\_df\_agecat)

summary(lm\_over50w)

decades <- seq(30, 60, by = 10)

over\_age\_m <- map\_dfc(decades, function(x) {

name <- paste0('over', x)

df <- data\_frame(pmax(0, men\_df\_agecat$age - x))

names(df) <- name

df

})

tail(over\_age\_m)

decades <- seq(30, 60, by = 10)

over\_age\_w <- map\_dfc(decades, function(x) {

name <- paste0('over', x)

df <- data\_frame(pmax(0, women\_df\_agecat$age - x))

names(df) <- name

df

})

tail(over\_age\_w)

lm\_piecewise\_m <- men\_df\_agecat %>%

bind\_cols(over\_age\_m) %>%

select(time, age, starts\_with('over')) %>%

lm(time ~ ., data = .)

summary(lm\_piecewise\_m)

lm\_piecewise\_w <- women\_df\_agecat %>%

bind\_cols(over\_age\_w) %>%

select(time, age, starts\_with('over')) %>%

lm(time ~ ., data = .)

summary(lm\_piecewise\_w)

over\_age\_df <- map\_dfc(decades, function(x) {

name <- paste0('over', x)

df <- data\_frame(pmax(0, age20to80 - x))

names(df) <- name

df

}) %>%

bind\_cols(age = age20to80, .)

tail(over\_age\_df)

predPiecewise\_m = predict(lm\_piecewise\_m, over\_age\_df)

predPiecewise\_w = predict(lm\_piecewise\_w, over\_age\_df)

plot(predPiecewise\_m ~ age20to80,

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

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

lines(x = age20to80, y = men\_res\_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")

dev.copy(png,filename="./cs2Figures/predPiecewise\_men.png", width=400, height=400);

dev.off ();

plot(predPiecewise\_w ~ age20to80,

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

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

lines(x = age20to80, y = women\_res\_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")

dev.copy(png,filename="./cs2Figures/predPiecewise\_women.png", width=400, height=400);

dev.off ();

men\_df\_agecat %>%

filter(year == 2012) %>%

select(time) %>%

summary()

women\_df\_agecat %>%

filter(year == 1999) %>%

select(time) %>%

summary()

age1999m = men\_df\_agecat[ men\_df\_agecat$year == 1999, "age" ]

age2012m = men\_df\_agecat[ men\_df\_agecat$year == 2012, "age" ]

age1999w = women\_df\_agecat[ men\_df\_agecat$year == 1999, "age" ]

age2012w = women\_df\_agecat[ men\_df\_agecat$year == 2012, "age" ]

plot(density(age1999m, na.rm = TRUE),

ylim = c(0, 0.075), col = "purple",

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

lines(density(age2012m, na.rm = TRUE),

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

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

legend = c("1999m", "2012m"), bty = "n")

dev.copy(png,filename="./cs2Figures/density\_men2\_1999\_2012.png", width=400, height=400);

dev.off ();

plot(density(age1999w, na.rm = TRUE),

ylim = c(0, 0.075), col = "purple",

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

lines(density(age2012w, na.rm = TRUE),

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

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

legend = c("1999w", "2012w"), bty = "n")

dev.copy(png,filename="./cs2Figures/density\_women2\_1999\_2012.png", width=400, height=400);

dev.off ();

## Loess curves fit to performance for 1999 and 2012 male runners

mR.lo99 = loess(time ~ age, men\_df\_agecat[ men\_df\_agecat$year == 1999,])

mR.lo.pr99 = predict(mR.lo99, data.frame(age = age20to80))

mR.lo12 = loess(time ~ age, men\_df\_agecat[ men\_df\_agecat$year == 2012,])

mR.lo.pr12 = predict(mR.lo12, data.frame(age = age20to80))

plot(mR.lo.pr99 ~ age20to80,

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

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

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("1999m", "2012m"), bty = "n")

dev.copy(png,filename="./cs2Figures/loess\_men\_1999\_2012.png", width=400, height=400);

dev.off ();

## Difference between Loess Curves

## This line plot shows the difference between the predicted run time for 2012 and 1999 male runners

lo\_pr\_99m <- predict(loess(time ~ age,

data = men\_df\_agecat,

subset = men\_df\_agecat$year == 1999),

data\_frame(age = age20to80))

lo\_pr\_12m <- predict(loess(time ~ age,

data = men\_df\_agecat,

subset = men\_df\_agecat$year == 2012),

data\_frame(age = age20to80))

data\_frame(age = age20to80, diff = lo\_pr\_12m - lo\_pr\_99m) %>%

ggplot(aes(age, diff)) +

geom\_line(color ='#3366FF') +

labs(x = 'Age', y = 'Difference in Fitted Curves (minutes)')

dev.copy(png,filename="./cs2Figures/loess\_men\_1999\_2012.png", width=400, height=400);

dev.off ();

## Loess curves fit to performance for 1999 and 2012 female runners

wR.lo99 = loess(time ~ age, women\_df\_agecat[ women\_df\_agecat$year == 1999,])

wR.lo.pr99 = predict(wR.lo99, data.frame(age = age20to80))

wR.lo12 = loess(time ~ age, women\_df\_agecat[ women\_df\_agecat$year == 2012,])

wR.lo.pr12 = predict(wR.lo12, data.frame(age = age20to80))

plot(wR.lo.pr99 ~ age20to80,

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

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

lines(x = age20to80, y = wR.lo.pr12,

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

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

legend = c("1999w", "2012w"), bty = "n")

dev.copy(png,filename="./cs2Figures/loess\_women\_1999\_2012.png", width=400, height=400);

dev.off ();

men\_res\_lo <- loess(time ~ age, data = men\_df\_agecat)

men\_res\_lo\_norm99 <- loess(Time ~ as.numeric(Age), data = men\_99\_norm)

men\_res\_lo\_norm12 <- loess(Time ~ as.numeric(Age), data = men\_12\_norm)

women\_res\_lo <- loess(time ~ age, data = women\_df\_agecat)

women\_res\_lo\_norm99 <- loess(Time ~ as.numeric(Age), data = women\_99\_norm)

women\_res\_lo\_norm12 <- loess(Time ~ as.numeric(Age), data = women\_12\_norm)

age20to80 <- 20:80

men\_res\_lo\_pr <- predict(men\_res\_lo, data.frame(age = age20to80))

men\_res\_lo\_pr\_norm99 <- predict(men\_res\_lo\_norm99)

men\_res\_lo\_pr\_norm12 <- predict(men\_res\_lo\_norm12)

women\_res\_lo\_pr <- predict(women\_res\_lo, data.frame(age = age20to80))

women\_res\_lo\_pr\_norm99 <- predict(women\_res\_lo\_norm99)

women\_res\_lo\_pr\_norm12 <- predict(women\_res\_lo\_norm12)

summary(men\_res\_lo\_pr\_norm99)

summary(men\_res\_lo\_pr\_norm12)

summary(women\_res\_lo\_pr\_norm99)

summary(women\_res\_lo\_pr\_norm12)

# Normalized loess predictions

plot(x = men\_12\_norm$Age, y = men\_res\_lo\_pr\_norm12,

col = "purple", lty = 2, lwd = 3,

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

lines(x = men\_99\_norm$Age, y = men\_res\_lo\_pr\_norm99,

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

legend("bottomleft", col = c("black", "purple"), lty = 1:2, lwd = 3,

legend = c("2012w", "1999w"), bty = "n")

dev.copy(png,filename="./cs2Figures/loessDiff\_men\_1999\_2012\_normalized.png", width=400, height=400);

dev.off ();

# Normalized loess predictions

plot(x = women\_12\_norm$Age, y = women\_res\_lo\_pr\_norm12,

col = "purple", lty = 2, lwd = 3,

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

lines(x = women\_99\_norm$Age, y = women\_res\_lo\_pr\_norm99,

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

legend("bottomleft", col = c("black", "purple"), lty = 1:3, lwd = 3,

legend = c("2012w", "1999w"), bty = "n")

dev.copy(png,filename="./cs2Figures/loessDiff\_women\_1999\_2012\_normalized.png", width=400, height=400);

dev.off ();

## Difference between Loess Curves

## This line plot shows the difference between the predicted run time for 2012 and 1999 female runners

lo\_pr\_99w <- predict(loess(time ~ age,

data = women\_df\_agecat,

subset = women\_df\_agecat$year == 1999),

data\_frame(age = age20to80))

lo\_pr\_12w <- predict(loess(time ~ age,

data = women\_df\_agecat,

subset = women\_df\_agecat$year == 2012),

data\_frame(age = age20to80))

data\_frame(age = age20to80, diff = lo\_pr\_12w - lo\_pr\_99w) %>%

ggplot(aes(age, diff)) +

geom\_line(color='#CC6666') +

labs(x = 'Age', y = 'Difference in Fitted Curves (minutes)')

dev.copy(png,filename="./cs2Figures/loessDiff\_women\_1999\_2012.png", width=400, height=400);

dev.off ();

## Section 6: Constructe a record for an individual runner across years

## How many entrants are there over the 14 years?

## How many unique names are there among these entrants?

## How many names appear twice, 3 times, 4 times, etc. and what name occurs most often?

## How often does a name appear more than once in a year?

# How many entrants are there over the 14 races?

length(men\_df\_agecat$name)

length(women\_df\_agecat$name)

# How many unique names are there?

length(unique(men\_df\_agecat$name))

length(unique(women\_df\_agecat$name))

# How many names appear once, twice, etc.?

table(table(men\_df\_agecat$name))

table(table(women\_df\_agecat$name))

# Which name appears 30 times?

head(sort(table(men\_df\_agecat$name), decreasing = TRUE), 1)

head(sort(table(women\_df\_agecat$name), decreasing = TRUE), 1)

# Let's examine other information about these 33 Michael Smiths

msmith <- men\_df\_agecat %>% filter(name == 'Michael Smith')

jjohnson <- women\_df\_agecat %>% filter(name == 'Jennifer Johnson')

# The hometowns

msmith %>% count(home) %>% arrange(desc(n)) %>% head()

jjohnson %>% count(home) %>% arrange(desc(n)) %>% head()

## Remove punctuation such as a period after someone's middle initial and any stray commas

men\_df\_agecat <- men\_df\_agecat %>%

mutate(name = str\_replace\_all(name, '[,.]', ''))

women\_df\_agecat <- women\_df\_agecat %>%

mutate(name = str\_replace\_all(name, '[,.]', ''))

## Figure out how many times a name appears in the same year

tab\_name\_yr\_m <- table(men\_df\_agecat$year, men\_df\_agecat$name)

tab\_name\_yr\_w <- table(women\_df\_agecat$year, women\_df\_agecat$name)

## Call 'max()' to find the cell in the table with the greatest count

max(tab\_name\_yr\_m)

max(tab\_name\_yr\_w)

### Check with class(), mode(), and names()

class(tab\_name\_yr\_m)

mode(tab\_name\_yr\_m)

names(attributes(tab\_name\_yr\_m))

class(tab\_name\_yr\_w)

mode(tab\_name\_yr\_w)

names(attributes(tab\_name\_yr\_w))

## Call dim() and colnames() to find the implications of this data structure

dim(tab\_name\_yr\_m)

head(colnames(tab\_name\_yr\_m), 3)

dim(tab\_name\_yr\_w)

head(colnames(tab\_name\_yr\_w), 3)

## Use which() to find the row and column location and need to include the 'arr.ind' argument in the call

idx\_max\_m <- which(tab\_name\_yr\_m == max(tab\_name\_yr\_m), arr.ind = TRUE)

idx\_max\_m

idx\_max\_w <- which(tab\_name\_yr\_w == max(tab\_name\_yr\_w), arr.ind = TRUE)

idx\_max\_w

## Locate the names (it is Michael Brown, not the Michael Smith)

colnames(tab\_name\_yr\_m)[idx\_max\_m[2]]

colnames(tab\_name\_yr\_w)[idx\_max\_w[2]]

## Create yob() in the data frame to drive an approximation to year of birth

men\_df\_agecat <- men\_df\_agecat %>%

mutate(yob = year - age)

women\_df\_agecat <- women\_df\_agecat %>%

mutate(yob = year - age)

## Check the values for these new and cleaned variables for 'Michael Brown'

men\_df\_agecat %>%

filter(name == 'Michael Brown') %>%

arrange(yob) %>%

select(year, name, home, yob, time)

women\_df\_agecat %>%

filter(name == 'Aarti Shah') %>%

arrange(yob) %>%

select(year, name, home, yob, time)

## Paste together the cleaned name and the derived year of birth

men\_df\_agecat <- men\_df\_agecat %>%

mutate(id = paste(name, yob, sep = '\_'))

## Determine how many times each 'id' appears in 'men\_df\_agecat'

races <- men\_df\_agecat %>% count(id)

## Select those IDs that appear at least 8 times

races8 <- races %>% filter(n >= 8) %>% .$id

## Subset 'men\_df\_agecat' to select the entries belonging to these identifiers

men8 <- men\_df\_agecat %>% filter(id %in% races8)

## Organize the data frame so that entries with the same 'id' are contiguous

men8 <- men8 %>% arrange(id)

## Alternatively, create a liost to store elements for each ID in 'races8'

men8L <- men8 %>% group\_by(id) %>% nest() %>% .$data

names(men8L) <- races8

## Check how many left IDs

length(unique(men8$id))

## Determine which satisfy 20 min constraint

gap\_time <- map\_lgl(men8L, ~ any(abs(diff(.$time)) > 20))

## How many of these runners have gaps of more than 20 minutes

sum(gap\_time)

## Slightly reformatted displays of the first two of these athletes are

map(men8L[gap\_time][1:2], ~ .[ ,c('year', 'home', 'name', 'yob', 'time')])

# gsub("[[:blank:]][a-z]{2}$", "", home)

# gsub("[[:blank:]][a-z]{2}$", "", home)

## Determine how many characters are in each value for home

home\_len\_m <- nchar(men\_df\_agecat$home)

home\_len\_w <- nchar(women\_df\_agecat$home)

## use it to extract the last two characters and add them back to our data frame

men\_df\_agecat <- men\_df\_agecat %>%

mutate(state = str\_sub(home, start = home\_len\_m - 1, end = home\_len\_m))

women\_df\_agecat <- women\_df\_agecat %>%

mutate(state = str\_sub(home, start = home\_len\_w - 1, end = home\_len\_w))

## Set the 2006 values to `NA`:

men\_df\_agecat$state[men\_df\_agecat$year == 2006] <- NA

women\_df\_agecat$statewo[women\_df\_agecat$year == 2006] <- NA

## Recreate the new 'id' so that it includes 'state'

men\_df\_agecat <- men\_df\_agecat %>%

mutate(id = paste(name, yob, state, sep = '\_'))

women\_df\_agecat <- women\_df\_agecat %>%

mutate(id = paste(name, yob, state, sep = '\_'))

## Again select those 'id's that occur at least 8 times

races\_m <- men\_df\_agecat %>% count(id)

races8\_m <- races\_m %>% filter(n >= 8) %>% .$id

men8 <- men\_df\_agecat %>% filter(id %in% races8\_m) %>% arrange(id)

men8L <- men8 %>% group\_by(id) %>% nest() %>% .$data

names(men8L) <- races8\_m

races\_w <- women\_df\_agecat %>% count(id)

races8\_w <- races\_w %>% filter(n >= 8) %>% .$id

women8 <- women\_df\_agecat %>% filter(id %in% races8\_w) %>% arrange(id)

women8L <- women8 %>% group\_by(id) %>% nest() %>% .$data

names(women8L) <- races8\_w

## length() for getting the number of runners who havea completed 8 races

length(races8\_m)

length(races8\_w)

## Section 7: Modelling

## Divide the runners into 9 groups to make 9 plots in a 3-by-3 grid

## Assign roughly the same number of runners to each group

men8 <- men8 %>%

mutate(group = group\_indices(., id) %% 9 + 1)

women8 <- women8 %>%

mutate(group = group\_indices(., id) %% 9 + 1)

## Plot it with one call to 'ggplot()' for men runners

ggplot(men8, aes(age, time, color = id)) +

geom\_line(show.legend = FALSE) +

facet\_wrap( ~ group)

dev.copy(png,filename="./cs2Figures/group9\_men\_1999\_2012.png", width=400, height=400);

dev.off ();

## Plot it with one call to 'ggplot()' for women runners

ggplot(women8, aes(age, time, color = id)) +

geom\_line(show.legend = FALSE) +

facet\_wrap( ~ group)

dev.copy(png,filename="./cs2Figures/group9\_women\_1999\_2012.png", width=400, height=400);

dev.off ();

## Draw a plot with the fitted lines to capture each runner's performance in group 9

men8 %>%

filter(group == 9) %>%

ggplot(aes(age, time, color = id)) +

geom\_line(show.legend = FALSE) +

geom\_smooth(aes(group = id),

method = 'lm',

se = FALSE,

color = 'grey50',

size = 0.5,

linetype = 'dashed',

show.legend = FALSE) +

facet\_wrap( ~ group)

dev.copy(png,filename="./cs2Figures/group9th\_men\_1999\_2012.png", width=400, height=400);

dev.off ();

## Draw a plot with the fitted lines to capture each runner's performance in group 9

women8 %>%

filter(group == 9) %>%

ggplot(aes(age, time, color = id)) +

geom\_line(show.legend = FALSE) +

geom\_smooth(aes(group = id),

method = 'lm',

se = FALSE,

color = 'grey50',

size = 0.5,

linetype = 'dashed',

show.legend = FALSE) +

facet\_wrap( ~ group)

dev.copy(png,filename="./cs2Figures/group9th\_women\_1999\_2012.png", width=400, height=400);

dev.off ();

## Examine the runner-to-runner variability, fit lines to all 306 athletes

lm\_coefs\_m <- men8 %>%

group\_by(id) %>%

do(fit = lm(time ~ age, data = .)) %>%

tidy(fit)

long\_coefs\_m <- men8 %>%

group\_by(id) %>%

summarise(med\_age = median(age)) %>%

mutate(coef = lm\_coefs\_m$estimate[lm\_coefs\_m$term == 'age'])

## Coefficients from Longitudinal Analysis of Athletes

## Scatter plot displays the slope of the fitted line to each of the 300+ runners who competed in at least 8 Cherry Blossom road races

## A negative coefficient indicates the runner is getting faster as he ages

## The plot includes a least squares fitted line and a loess fitted curve

ggplot(long\_coefs\_m, aes(med\_age, coef)) +

geom\_point(size = 0.7) +

geom\_smooth(method = 'lm', size = 0.7, color = 'darkgreen', se = FALSE) +

geom\_smooth(method = 'loess', size = 0.7, linetype = 'dashed', se = FALSE) +

geom\_hline(yintercept = 0, color = 'grey70', size = 0.3) +

labs(x = 'Median age', y = 'Coefficient (minutes per race / year)')

dev.copy(png,filename="./cs2Figures/coef\_variability\_men.png", width=400, height=400);

dev.off ();

## How these coefficients vary with age?

long\_fit\_m <- lm(coef ~ med\_age, data = long\_coefs\_m)

summary(long\_fit\_m)

## Examine the runner-to-runner variability, fit lines to all 306 athletes

lm\_coefs\_w <- women8 %>%

group\_by(id) %>%

do(fit = lm(time ~ age, data = .)) %>%

tidy(fit)

long\_coefs\_w <- women8 %>%

group\_by(id) %>%

summarise(med\_age = median(age)) %>%

mutate(coef = lm\_coefs\_w$estimate[lm\_coefs\_w$term == 'age'])

## Coefficients from Longitudinal Analysis of Athletes

## Scatter plot displays the slope of the fitted line to each of the 300+ runners who competed in at least 8 Cherry Blossom road races

## A negative coefficient indicates the runner is getting faster as he ages

## The plot includes a least squares fitted line and a loess fitted curve

ggplot(long\_coefs\_w, aes(med\_age, coef)) +

geom\_point(size = 0.7) +

geom\_smooth(method = 'lm', size = 0.7, color = 'darkgreen', se = FALSE) +

geom\_smooth(method = 'loess', size = 0.7, color = '#CC6666', linetype = 'dashed', se = FALSE) +

geom\_hline(yintercept = 0, color = 'grey70', size = 0.3) +

labs(x = 'Median age', y = 'Coefficient (minutes per race / year)')

dev.copy(png,filename="./cs2Figures/coef\_variability\_women.png", width=400, height=400);

dev.off ();

## How these coefficients vary with age?

long\_fit\_w <- lm(coef ~ med\_age, data = long\_coefs\_w)

summary(long\_fit\_w)

## -- The end of codes for case study 2 --

## **Appendix – Figures S1-5**

|  |  |
| --- | --- |
| **Figure S1: Bar plots to show the number of male and female runners by year**  These plots show that the numbers of male (**Figure S1A**) and female (**Figure S1B**) runners in the Cherry Blossom 10-mile race has more than doubled from 1999 to 2012. | |
| **A: Male** | **B: Female** |

|  |  |
| --- | --- |
| **Fig. S2: Residual plots from fitting simple linear models of performance to age for runners**  The smoothed scatter plots of the residuals from the fit of the simple linear model of run time to age for male (**Figure S2A**) and female (**Figure S2B**) runners. Two overlaid curves: a solid horizontal line at y = 0 (solid purple) and a local smooth of the residuals (dashed green). | |
| **A: Male** | **B: Female** |
|  |  |

|  |  |
| --- | --- |
| **Fig.S3: Run times for multiple races**  These line plots show the times for male (**FigureS3A**) and female (**Figure S3B**) runners who completed at least 8 Cherry Blossom races. Each set of connected segments corresponds to the run times for one athlete. Similar changes are seen in the scatter plots and the changes for individuals. For example, many middle-aged runners show a sharp increase in run time with age but that is not the case for all. Some of them improve and others change more slowly. | |
| **A: Male** | **B: Female** |
| **Fig. S4: Linear fits of run time to age for individual runners**  Here we have augmented the bottom-right line plot from **Figure S3** with the least squares fit of run time for each of the male (**Figure S4A**) and female (**FigureS4B**) athletes. These are the 30 or so black dashed line segments plotted on each of the individual runner’s times series. | |
| **A: Male** | **B: Female** |
| **Fig. S5: Coefficients from Longitudinal Analysis of Athletes**  This scatter plot displays the slope of the fitted line to each of the 300+ runners who competed in at least 8 Cherry Blossom road races. A negative coefficient indicates the runner is getting faster as he ages. The plot includes a least square fitted line and a loess fitted curve. Notice that nearly all of the coefficients for those over 50 are positive. The typical size of this coefficient for a 50-year old is about one minute per year. | |
| **A: Male** | **B: Female** |