Modeling Runners' Times in the Cherry Blossom Race

Justin Howard, Stuart Miller, Paul Adams

September 22, 2020

1 Introduction

The internet is a vast open resource for data, but internet data can be messy and difficult to wrangle. In this case study we collect data on participates of the Cherry Blossom Race from the Cherry Blossom Race website[1]. We find that there are a number of issues present in the data, including inconssitent web addressing formatting, inconsistent formatting, and missing data. We collect and clean this data to provide the Men's and Women's information in a format suitable for analysis.

2 Methods

2.1 Data

2.1.1 Data Collection

In this case study, we collected data on participants in the Cherry Blossom Race. We scraped the data for Men and Woman from 1999 to 2012. The data was stored under the base address in the following form, where the year is between 1999 and 2012 and page is a specific page name.

`http://www.cherryblossom.org/results/<year>/<page>`

The specific addresses where the data is stored are given in table 1. While the structure follows `results/<year>` consistently, the page naming for each year is not consistent. In some cases, the names for the men's and women's pages are `men` and `women`, repectively. In other cases, a code is used such as `cb99m` and `cb99f` in 1999 for `men` and `women` repectively.

[1]See http://www.cherryblossom.org/.

Table 1. Web Page Pages

Year	Men's Pages	Women's Pages
1999	`results/1999/cb99m.html`	`results/1999/cb99f.html`
2000	`results/2000/Cb003m.htm`	`results/2000/Cb003f.htm`
2001	`results/2001/oof_m.html`	`results/2001/oof_f.html`
2002	`results/2002/oofm.htm`	`results/2002/ooff.htm`
2003	`results/2003/CB03-M.HTM`	`results/2003/CB03-F.HTM`
2004	`results/2004/men.htm`	`results/2004/women.htm`
2005	`results/2005/CB05-M.htm`	`results/2005/CB05-F.htm`
2006	`results/2006/men.htm`	`results/2006/women.htm`
2007	`results/2007/men.htm`	`results/2007/women.ht`
2008	`results/2008/men.htm`	`results/2008/women.htm`
2009	`results/2009/09cucb-M.htm`	`results/2009/09cucb-F.htm`
2010	`results/2010/2010cucb10m-m.htm`	`results/2010/2010cucb10m-F.htm`
2011	`results/2011/2011cucb10m-m.htm`	`results/2011/2011cucb10m-F.htm`
2012	`results/2012/2012cucb10m-m.htm`	`results/2012/2012cucb10m-F.htm`

The data are successfully scraped from the urls that are given and we observe that the number of participants in the Cherry Blossom race has increased steadily from 1999 to 2012. We can use the `extractResTable` function to exptract all of the results, or to specify a single set of results.

We observe the formatting of the text data, such as the use of the `=` character can be used to transform the lines of text into a a matrix. We will used this pattern to identify the boundary between the headering and the body of the data. Once we defined the column names, we were free to use the text structure, such as the spacing of column names and data, to identify datapoints. We started with age, which denoted by the `"ag"` identifier. After we sucessfully used the structure of the text data to identify an index location for `"ag"` data points. We expanded the search to inlcude the spacing of text data to identify other data points. The spacing indices were used in a funtion that compiled a matrix of data points across the entire list of tables.

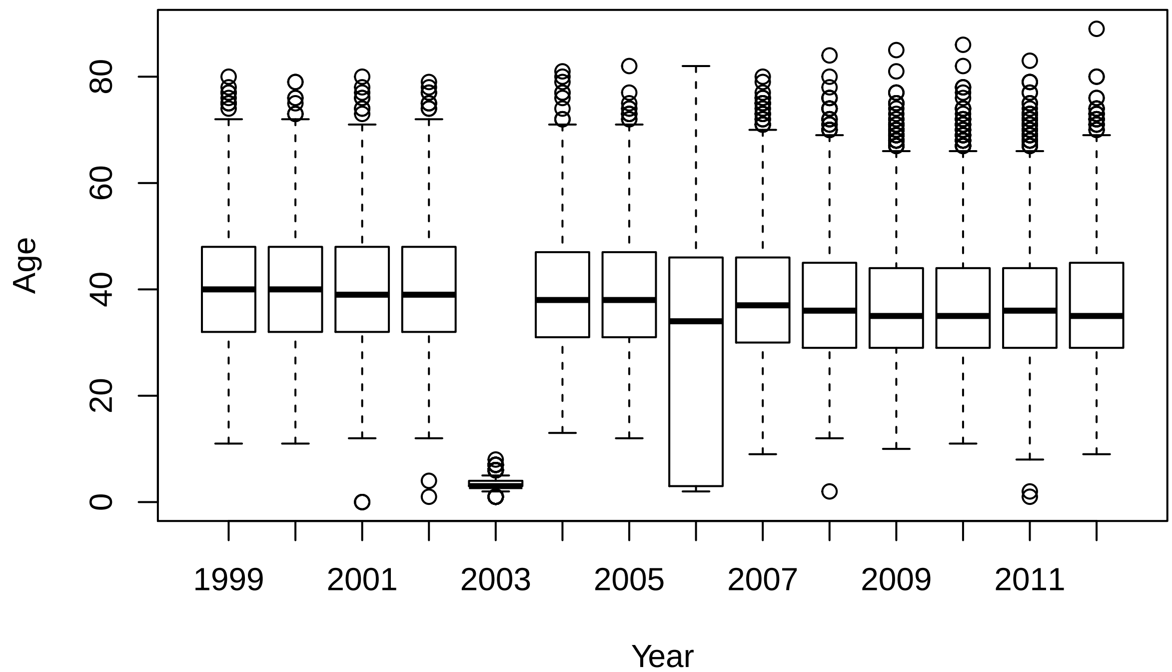
A function `findColLocs` was defined to use these space indices to locate the begin points for data throughout the extracted text data. `selectCols` is a tool that uses `findColLocs` to locate the data points for each column throughout the text for all of the race years. Our functions extracted almost all of the age data fomr the text, which aided in the identification errors within the dataset. To generalize this method, we defined a function, `extractVariables`, that combined the various processes we used to preprocess the data into a more machine readable format.

We now have a list of matrices. The original text data are in a much more machine readable format. Our number of rows for each matrix is equal to the number of original rows, minus the original text header. To perform statistical analyses on the numeric data, we must perform a datatype transformation.

There were anomalies in the data for the year 2003. After an examination of the original data table, we found that the spacing in the year 2003 data is not the same as the other years. We adjusted our `selectCols` function to better capture the data for 2003. We also observe anomalies in the formatting of the year 2001. We introduced fixes and re-scraped the data once more to form the list of matrices called `menResMat`.

" 1 1/1420 1 Ridouane Harroufi 27 Morocco 45:56 45:56# 4:36 "

" 27 " " " " " " " 26 M" " 26 " " 23 Ken"



After these anomalies were corrected, we reformatted the datatypes. The `"ag"` data was changed to a numeric format for analysis, which revealed *more* anomalies. We observed suspiciously young ages between the years 2001 and 2003. These ages were revealed to be *erroneous* upon closer examination. These values were removed from the dataset.

To make the remaining analyses easier, we transformed the data structure from a list of matrices into a dataframe. Combining the data into a single dataframe facilitated the cleaning of the columns containing the time data for each runner. These values were stored as characters and were in an HR:MIN:SEC format. We can separate the values and convert them all into a standard time format, minutes. We defined a function that converts the time into a minute format.

The end result was a dataframe called menDF. With this process complete, we simply re-used the functions on the women's data.

In preparation for further analysis, the values for 'home' needed to be updated. First, 12 runners were identified as having at least one iteration of a bad address. Based on state codes provided with those addresses, alternative addresses used by those runners in other races were applied in place of the bad addresses.

Additionally, because most of the addresses were not in standard format, we converted home names to ISO standards for national naming. For example, we converted 'Rep Of S.africa' to the iso3166-recognized 'South

Africa'. The names were originally given ISO Name values for analysis and reporting, then converted to ISO 3-byte country codes for cross-walking into the map's visual abstraction. All runners with specified nations (whether by ISO code or name) were listed with their nation while all others were listed as 'United States.' For cross-walking locations to states, we identified 7,398 records where a state was given, ending in value such as ' AK','‚AK', where 'AK' is 'Alaska'. This information was processed in creating Fig. 2 and Fig. 3.

Our dataset is complete, clean, and ready for further analysis.

```
##   year sex                   name                home age  runTime
## 1 1999   M Worku Bikila          Ethiopia             28 46.98333
## 2 1999   M Lazarus Nyakeraka     Kenya                24 47.01667
## 3 1999   M James Kariuki         Kenya                27 47.05000
## 4 1999   M William Kiptum        Kenya                28 47.11667
## 5 1999   M Joseph Kimani         Kenya                26 47.51667
## 6 1999   M Josphat Machuka       Kenya                25 47.55000
```

3 Results

As visualized in Fig. 2, Results from analyzing geographic runner information identified that the nations with the fastest run times, represented by average, were from Tanzania, Morocco, Ethiopia, Kenya and Mexico. The nations with the slowest average run times were Ireland, Slovenia, Lebanon, Republic of Korea, and Jordan.

Top 5 Fastest Nations	Overall Race Completion Time
Tanzania	46.067
Morocco	46.817
Ethiopia	46.983
Kenya	47.017
Mexico	47.317

Top 5 Slowest Nations	Overall Race Completion Time
Ireland	121.42
Slovenia	108.25
Lebanon	103.95
Republic of Korea	100.733
Jordan	100.067

Average Race Completion Time, by Country

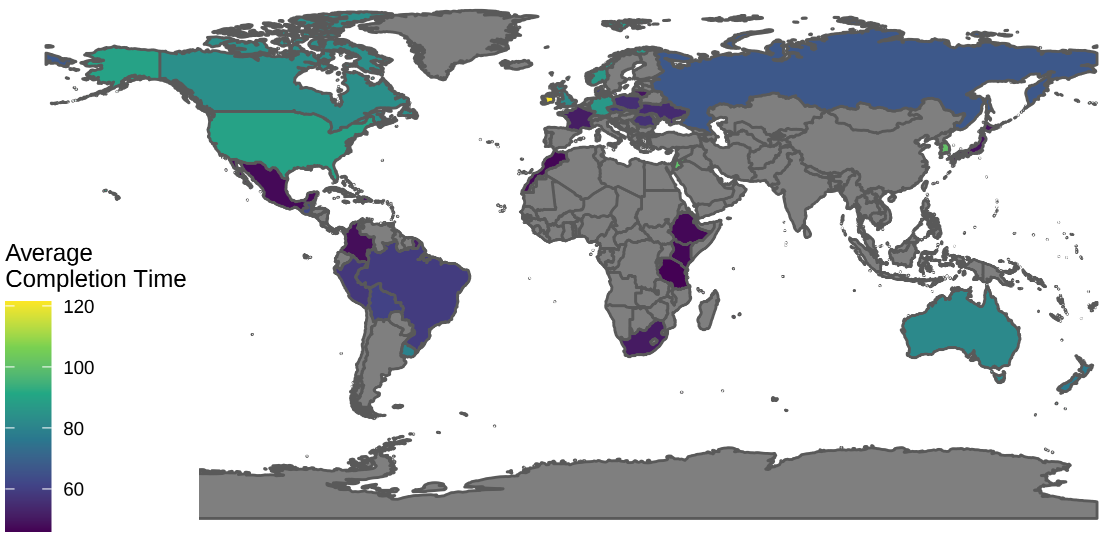


Figure 2:
Global Race Completion Times

As visualized in Fig. 3, the states represented by the fastest overall runners - based on average race completion time - are Nevada, Colorado, Delaware, Idaho, and New Mexico. The states with the slowest average race completion times are North Dakota, Kentucky, Hawaii, Oklahoma, and Oregon.

Top 5 Fastest States	Overall Race Completion Time
Nevada	76.269
Colorado	82.400
Delaware	83.019
Idaho	83.157
New Mexico	83.208

Top 5 Slowest States	Overall Race Completion Time
North Dakota	109.517
Kentucky	98.602
Hawaii	98.467
Oklahoma	97.837
Oregon	97.760

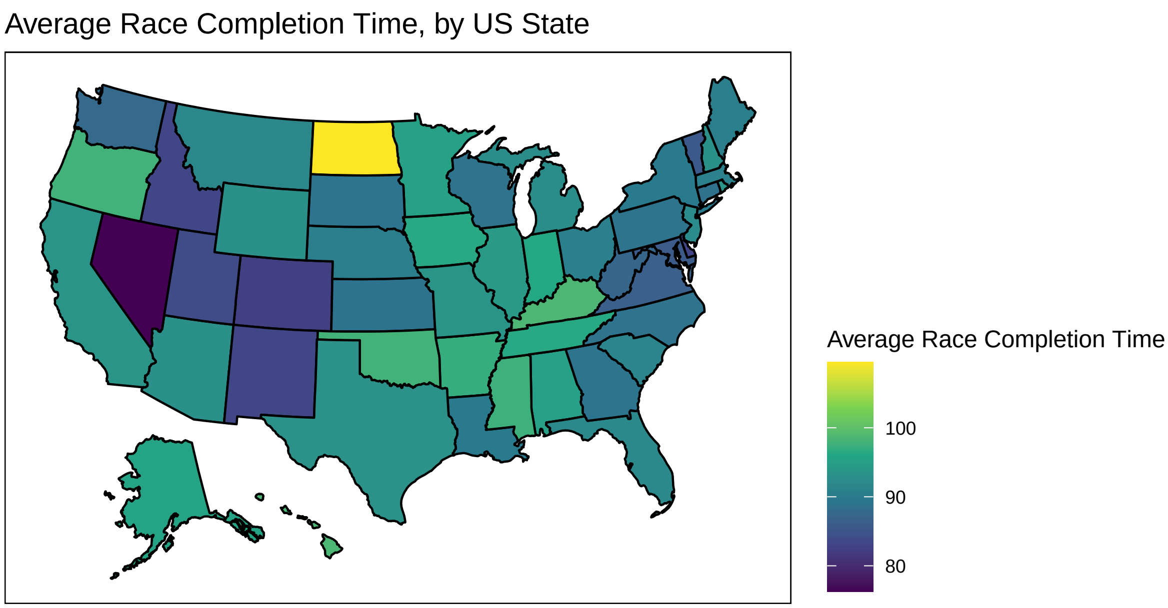


Figure 3:
Statewide Race Completion Times

4 Conclusion

These web-scraping methods proved useful for identifying the times of race completion per runner from year 1999 through 2012. From this information we were able to identify average race completion times, by nation and by state. However, documentation methodology and format varied over the years, which presented issues where data was required to be dropped - often as a result of missing information or incorrect information, such as children as young as 1 competing in the race - which impacted the results we were able to identify. National information was relatively reliable, but the state-level information was heavily under-represented. Because of the level of missing, unrecoverable information, we did not feel statistical analysis would have been useful. Many issues identified would have resulted in over-fitting during predicting modeling or time series (change-points) analysis.

It is important to note that there was perhaps ("perhaps" since we could not confirm exact locations) significant overlap between cities in Canada and the United States where nation and state were not provided. Therefore, race completion times between these two locations may not show as much variation as existed.

A Code

```
The code is cooool
```