DS 7333 - Quantifying the World

Case Study #2 - Modeling Runners' Times in the Cherry Blossom Race

Sterling Beason, Sean Kennedy, Emil Ramos

Introduction

Every year, the Credit Union Cherry Blossom Ten Mile Run takes place in Washington, D.C. and brings together credit unions from across the country with a a goal of fund raising for helping children. The proceeds from the race helps support Children’s Hospitals that belong to the Children's Miracle Network Hospitals, a non-profit international organization that helps to treat millions of children across the U.S. and Canada. The Ten Mile Run has grown in popularity and grown to as many as 17,000 runners with ages rangin from 9 to 89 participated.

In this case study, we are interested in understanding how people's physical performance changes as they age. We will use the information the race organizers collect about each runner which they publish individual-level data on their website. Some of the data includes runners' times and can provide us with insights to our question regarding performance and age.

Objective

Web scrape cherryblossom.org race results to seek out data to bring insight for female runners from 1999 to 2012 and explore the data.

Show Code

Download Copies of Pages

Code for fetching web based data

Read HTML into Memory

Persistence to reduce web calls

Extract Relevant Data

- Iterate through each page's HTML text
- Stripped all tags except "pre" with 'bleach' (this fixed HTML errors bs4 couldn't handle)
- Extract text contents of "pre" tag
- Split text by lines "\n"
- By line, seperated values between variable length of spaces
- Due to missing values on some pages, loop through the extracted values to find the most likely string to be a name. Then I assume the next value is an age. Age is checked to be a digit, else replaced with "N/A"
- At this stage, there are a couple of generic error messages used to track empty rows or no values matching a name. Those rows are skipped, but so far that looks acceptable.
- Each years rows are added to a single list. [[year, place, name, age]] (this can be changed to a dictionary by years if necessary.
- Data into a pandas DataFrame with columns. df

```
Processing year: 1999
==========================
empty row? (0)
 - row value: ['']
empty row? (2357)
 - row value: []
Processing year: 2000
==========================
empty row? (0)
 - row value: ['']
empty row? (2168)
 - row value: []
Processing year: 2001
==========================
empty row? (0)
 - row value: ['']
empty row? (1)
 - row value: ['']
Processing year: 2002
==========================
empty row? (0)
 - row value: ['']
empty row? (3336)
 - row value: []
Processing year: 2003
==========================
empty row? (0)
 - row value: ['']
empty row? (3544)
 - row value: ['Under USATF Age-Group']
empty row? (3545)
 - row value: []
Processing year: 2004
==========================
empty row? (0)
 - row value: ['']
empty row? (3901)
 - row value: ['Under USATF Age-Group']
Processing year: 2005
==========================
empty row? (0)
 - row value: ['']
empty row? (4334)
 - row value: ['']
empty row? (4336)
 - row value: ['Under USATF Age-Group']
Processing year: 2006
==========================
empty row? (0)
 - row value: ['']
empty row? (5437)
 - row value: ['Under USATF Age-Group']
empty row? (5438)
 - row value: []
Processing year: 2007
==========================
empty row? (0)
 - row value: ['']
empty row? (5692)
 - row value: ['Under USATF Age-Group']
empty row? (5693)
 - row value: []
Processing year: 2008
==========================
empty row? (0)
 - row value: ['']
empty row? (6398)
 - row value: []
Processing year: 2009
==========================
empty row? (0)
 - row value: ['']
empty row? (8325)
 - row value: ['Under USATF Age-Group']
empty row? (8326)
 - row value: []
Processing year: 2010
==========================
empty row? (0)
 - row value: ['']
empty row? (8855)
 - row value: ['Under USATF Age-Group']
empty row? (8856)
 - row value: []
Processing year: 2011
==========================
empty row? (0)
 - row value: ['']
empty row? (9031)
 - row value: []
Processing year: 2012
==========================
empty row? (0)
 - row value: ['']
empty row? (9730)
 - row value: []
```

Explore Data

Out[81]:

	year	place	name	age
0	1999	1	Jane Omoro	26.0
1	1999	2	Jane Ngotho	29.0
2	1999	3	Lidiya Grigoryeva	NaN
3	1999	4	Eunice Sagero	20.0
4	1999	5	Alla Zhilyayeva	29.0
...	...	...	...	...
76064	2012	9726	Khristina Nava	40.0
76065	2012	9727	Geneva Dixon	31.0
76066	2012	9728	Veronica Eligan	55.0
76067	2012	9729	Denise Bobba	40.0
76068	2012	9730	Rashonna Waples	38.0

76069 rows × 4 columns

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 76069 entries, 0 to 76068
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   year    76069 non-null  int64  
 1   place   76069 non-null  object 
 2   name    76069 non-null  object 
 3   age     76018 non-null  float64
dtypes: float64(1), int64(1), object(2)
memory usage: 2.3+ MB
```

```
51
```

Out[83]:

	year	place	name	age
2	1999	3	Lidiya Grigoryeva	NaN
7	1999	8	Gladys Asiba	NaN
16	1999	17	Connie Buckwalter	NaN
2174	1999	2175	Ann Reid	NaN
4522	2000	Under USATF OPEN guideline	Under USATF OPEN guideline	NaN
6954	2001	2432	Maria	NaN
7495	2001	Under USATF OPEN guideline	Under USATF OPEN guideline	NaN
7765	2002	270	Unknown RUNNER	NaN
8776	2002	1281	Melissa AKEY	NaN
9679	2002	2184	Yvonne BONNER	NaN
10245	2002	2750	Sylvia Susan DE LA	NaN
10434	2002	2939	Lori	NaN
10562	2002	3067	Mary	NaN
10756	2002	3261	XXXXX Unnamed Athlete	NaN
10830	2002	Under USATF OPEN guideline	Under USATF OPEN guideline	NaN
13734	2003	2904	Diem-Phuong	NaN
14373	2003	Under USATF OPEN guideline	Under USATF OPEN guideline	NaN
18273	2004	Under USATF OPEN guideline	Under USATF OPEN guideline	NaN
18336	2005	63	Lindsay Vogtsberger	NaN
18442	2005	169	Ashley Griffin	NaN
18760	2005	487	Angelica Jimenez	NaN
19048	2005	775	Runner I Iv Vii	NaN
19188	2005	915	Runner Xxxii	NaN
20646	2005	2373	Xandra Brandon	NaN
20674	2005	2401	Michelle Hinman	NaN
21231	2005	2958	Jennifer	NaN
21360	2005	3087	Michelle Merola	NaN
22109	2005	3836	Gwen	NaN
22172	2005	3899	Nancy Samko	NaN
22607	2005	Under USATF OPEN guideline	Under USATF OPEN guideline	NaN
26462	2006	3855	Robin Hershey	NaN
28043	2006	Under USATF OPEN guideline	Under USATF OPEN guideline	NaN
28389	2007	346	Chris Mickeever	NaN
29203	2007	1160	Doris Steere	NaN

	year	place	name	age
30014	2007	1971	Kimberly	NaN
33734	2007	Under USATF OPEN guideline	Under USATF OPEN guideline	NaN
38873	2008	5139	Maria De La Paz	NaN
40140	2009	9	Aziza Ayilu	NaN
43036	2009	2905	Maria De La Paz	NaN
45054	2009	4923	Katherine	NaN
47704	2009	7573	Maria Nelson	NaN
48455	2009	Under USATF OPEN guideline	Under USATF OPEN guideline	NaN
48693	2010	238	Lyly	NaN
50172	2010	1717	Maria De La Paz	NaN
55255	2010	6800	Tory Wilde	NaN
57309	2010	Under USATF OPEN guideline	Under USATF OPEN guideline	NaN
63337	2011	6028	Katherine	NaN
64021	2011	6712	Nancy	NaN
66184	2011	8875	Margret	NaN
66929	2012	590	Julita/	NaN
74586	2012	8248	Cynthia	NaN

Drop NAs and other Garbage Data

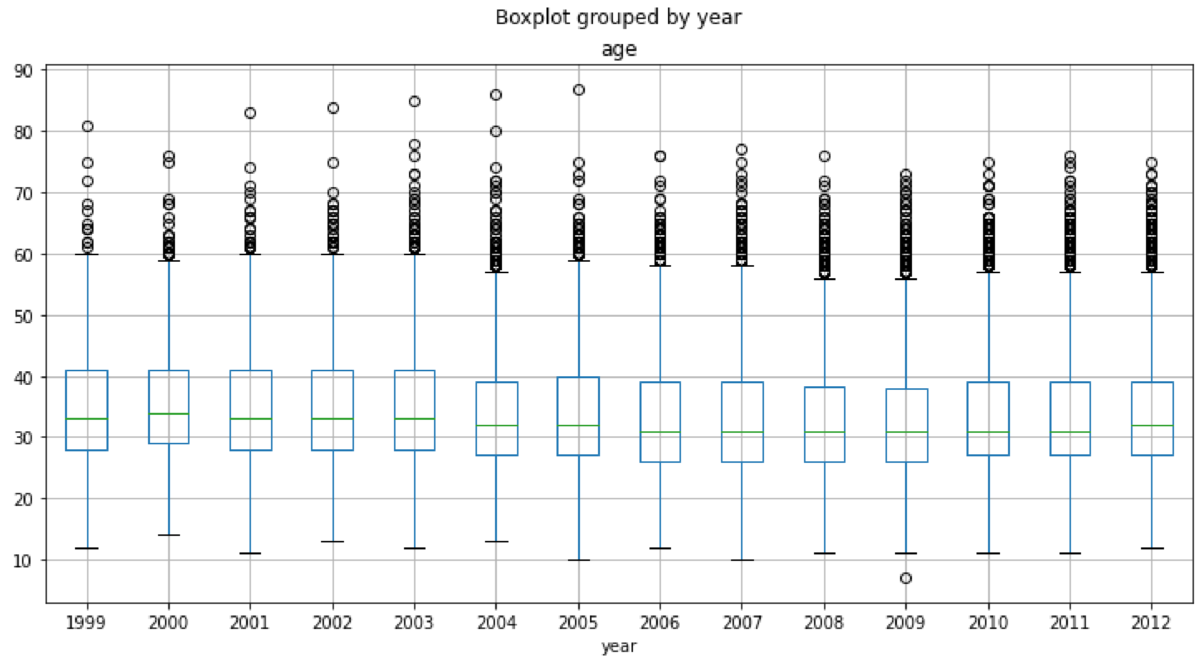
```
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-pa
ckages/pandas/core/frame.py:3997: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  errors=errors,
```

```
Out[84]: (76017, 4)
```

Question 10

We have seen that the 1999 runners were typically older than the 2012 runners. Compare the age distribution of the [female] runners across all 14 years of the races. Use quantile–quantile plots, boxplots, and density curves to make your comparisons. How do the distributions change over the years? Was it a gradual change?

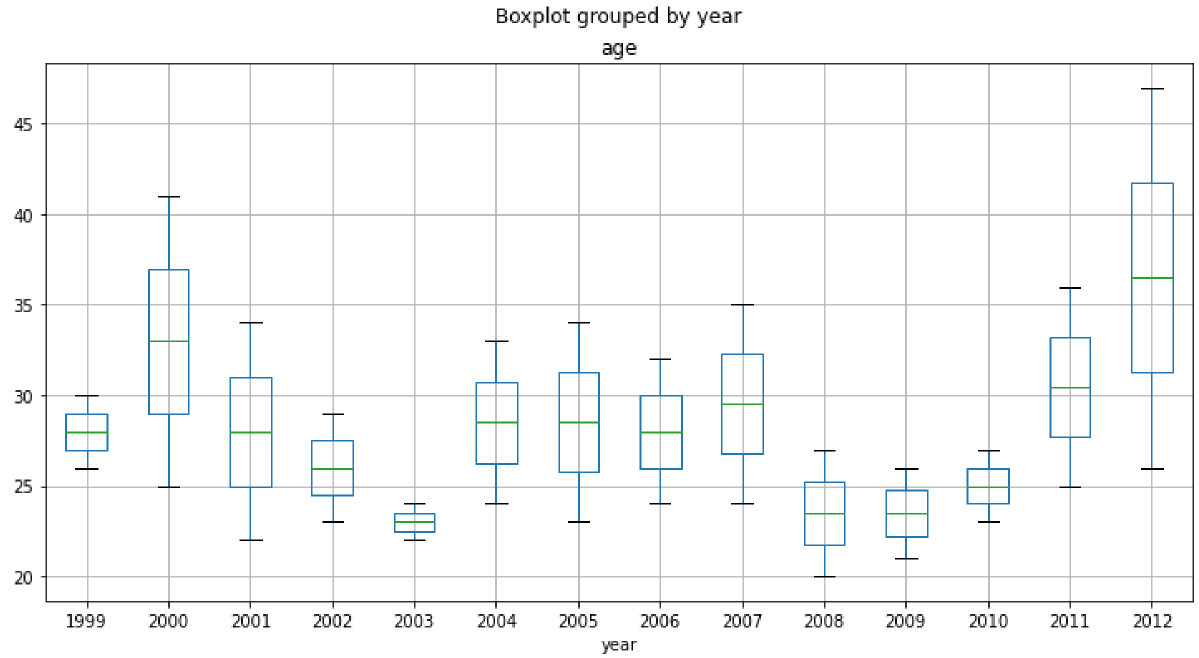
```
Out[85]: <matplotlib.axes._subplots.AxesSubplot at 0x114e8ceb8>
```



Analysis

Over the course of time the average age of players has remained relatively constant. Very few years had average ages above 35. But in order to establish whether the age of players that are actually good at tennis, not just able to play tennis, is changing as a function of time - we should limit our analysis to those players that finished in the top 10 or better.

```
Out[86]: <matplotlib.axes._subplots.AxesSubplot at 0x113b164a8>
```



Running the analysis in this way, we can see that the distrubution of ages has varied greatly over time - with the most recent year, 2012, having a wide-tailed distribution of ages from just over 25 to just under 50. Other eyars have had much tighter distributions for the top 10. Most notably the period from 2008 - 2010 where the average age in the top 10 was less than 25 and maxed out around 28. The periods of 2004-2007 saw a much higher average age in the top 10 than was show in subsequent years. This could be indicative of one older player occupying the top 10 during that period and the subsequently retiring in 2008 - let's check the data first.

Out[87]:

	year	place	name	age
22609	2006	2	Alevtina Ivanova	30.0

Out[88]:

	year	place	name	age
18275	2005	2	Alvetina Ivanova	29.0

Same player - let's map

```
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-pa
ckages/pandas/core/indexing.py:966: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  self.obj[item] = s

/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-pa
ckages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  """Entry point for launching an IPython kernel.
```

Out[91]:

	place	age
name		
Alevtina Ivanova	2.0	29.5
Atalelech Ketema	9.0	21.0
Aurica Buia	6.0	34.5
Aziza Aliyu	9.0	21.0
Catherine Ndereba	2.0	31.0
Dorota Gruca	7.0	36.0
Florence Jepkosgei	8.0	23.0
Galina Alexandrova	6.0	31.0
Ilona Barvanova	6.0	33.0
Isabella Ochichi	1.0	24.0
Kathy Butler	3.0	33.0
Kristin Price	10.0	24.0
Lidia Simon	4.0	32.5
Lidiya Grigoryeva	1.0	32.0
Lioudmila Kortchaguina	8.0	33.5
Lucinda Hull	8.0	24.0
Luminita Talpos	6.0	32.5
Magdalene Makunzi	4.0	24.0
Mary Kate Bailey	9.0	31.0
Naomi Wangui	7.0	25.0
Nicole Kulikov Hagobia	7.0	30.0
Nuta Olaru	1.0	34.0
Olga Romanova	5.0	23.0
Renata Paradowska	10.0	33.0
Sally Barsosio	5.0	27.0
Samia Akbar	10.0	23.0
Tatyana Petrova	3.0	21.5
Tetyana Hladyr	3.0	29.0
Teyba Erkesso	1.0	24.0
Turena Johnson Lane	5.0	30.0
Turena M Johnson Lane	6.0	29.0
Victoria Klimina	3.0	28.0

Aveerage age for players in the top 10 during this period was almost 28.5 years of age.

2008 - 2010

```
Out[92]: 28.34375
```

Out[94]:

	place	age
name		
Abebu Gelan	4.0	19.000000
Alemtshay Misganaw	6.0	29.000000
Alevtina Biktimirova	10.0	26.000000
Angelina Mutuku	2.0	25.000000
Aziza Aliyu	6.0	22.000000
Belainesh Zemedkun	3.0	22.000000
Belianesh Zemed Gebre	2.0	21.000000
Catherine Ndereba	4.5	35.500000
Claire Hallissey	8.0	27.000000
Hirut Mandefro	9.0	24.000000
Julliah Tinega	2.0	24.000000
Leah Kiprono	10.0	27.000000
Lidia Simon	7.5	34.500000
Lineth Chepkurui	1.0	21.333333
Misker Demessie	4.0	23.000000
Neriah Asiba	7.5	28.500000
Olga Romanova	6.0	28.000000
Phebe Ko	10.0	27.000000
Rashida Khayrutdinova	9.0	32.000000
Sally Meyerhoff	7.0	25.000000
Samia Akbar	7.0	28.000000
Sharon Cherop	5.0	24.000000
Tatyana Chulakh	8.0	25.000000
Teyba Naser	10.0	21.000000

```
Out[95]: 25.784722222222218
```

Data from 2008 - 2010 shows a significantly different trend towards younger players with an average age of 25.8 in the top 10.