Data story - 100 meters

Adrian Meier

February 7, 2021

Choice of technology

During the first interview David mentioned that the internship project is currently based on R. Having no prior experience with R I decided to take the learning opportunity and got to work with R and RMarkdown employing the libraries: rvest, dplyr, stringr, ggplot2, lubridate and ExtDist.

Choosing a starting point

The 100 meters Wikipedia page https://en.wikipedia.org/wiki/100_metres is full of data tables. A total of 17 tables rank racers in all kinds of categories. The ‘100 meters per age category’ section caught my eye. It lists the best performances of athletes between 5 and 19 years of age. What sparked my interest is the ‘Age’ column. The record holders aged five are both less than two weeks away from their sixth birthday. Clearly the age has a significant impact on performance at an early age. As a first question, I want to investigate the correlation of the athletes age (within an age category) to their absolute age.

Scraping wikipedia

Lets begin by reading the two tables of the *100 meters per age category* section. I wrote a function using the rvest library to read a HTML table given the corresponding XPATH from the wiki URL once. The table is stored locally as a csv file and loaded locally if available. Then I call the function with the arguments for the tables of interest.

```r
# load data from file directly if already scraped data and scrape and write csv file otherwise
load_data = function(data_name, data_xpath){
  data_filepath = paste("~/sprint/data/", data_name, ".csv", sep = "")
  # check if data file exists
  if (!file.exists(data_filepath)){
    # access data at url by XPATH
    loaded_data = url %>%
      read_html() %>%
      html_node(xpath = data_xpath) %>%
      html_table(fill = TRUE)
    # export to csv
    write.csv(loaded_data, file=data_filepath, row.names = FALSE)
  } else {
    loaded_data = read.csv(data_filepath, check.names = FALSE)
  }
  return(loaded_data)
}
```

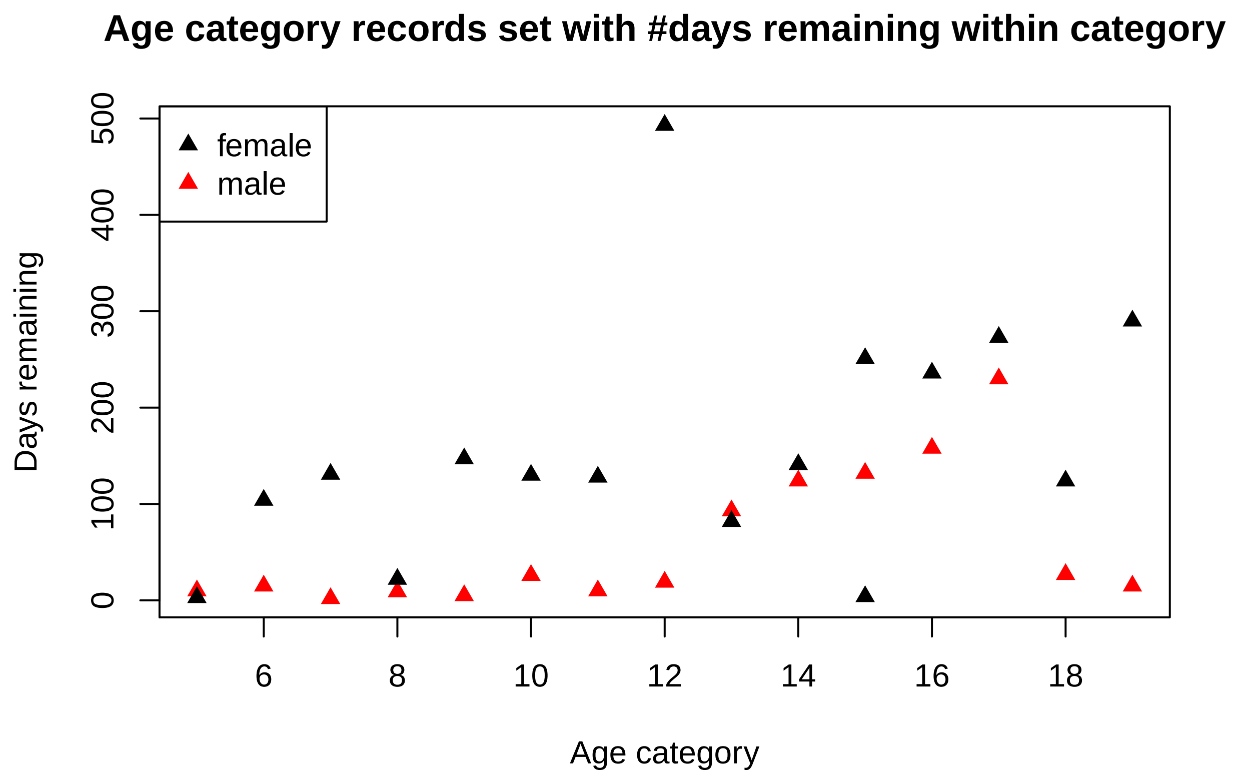
```
## Observations: 16
## Variables: 7
## $ Age          <int> 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 15, 16, 17...
## $ Time         <dbl> 16.12, 14.89, 13.97, 13.55, 12.67, 12.15, 11.75, ...
## $ `Wind (m/s)` <dbl> 1.6, 0.0, -0.4, 1.5, 1.7, 0.5, 1.6, 1.6, -1.2, 1....
## $ Athlete      <fct> Micahlena Cotton, Stacey Onyepunuka, Payton Payne...
## $ Date         <fct> 9 July 2016, 6 July 2013, 25 July 2015, 1 June 20...
## $ Place        <fct> "Orlando, United States", "Mesa, United States", ...
## $ Age          <fct> "5 years, 362 days", "6 years, 261 days", "7 year...
```

Next I do some post processing on the tables. To prevent ambiguity I rename the first 'Age' column to 'Age_category'. Further I drop columns that I won't need, add a gender flag and then combine the data in a single table.

```
## Observations: 31
## Variables: 4
## $ Age_category <int> 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18...
## $ Athlete      <fct> Kai Sapp, Willie Washington, Willie Washington, W...
## $ Age          <fct> "5 years, 355 days", "6 years, 350 days", "7 year...
## $ Gender       <chr> "male", "male", "male", "male", "male", "male", "...
```

Now I add a column which will store the days until the athletes birthday. This shows how many (or few) days were remaining for the athlete to compete in this age category (and potentially get a chance to improve their time). To calculate the remaining days, I have to compare the age (given as "X years, Y days") with the age category. Maybe there are neat functions which can handle this data format… I solved it with some stringr regexp magic.

Having done that, it's time for the first plot!

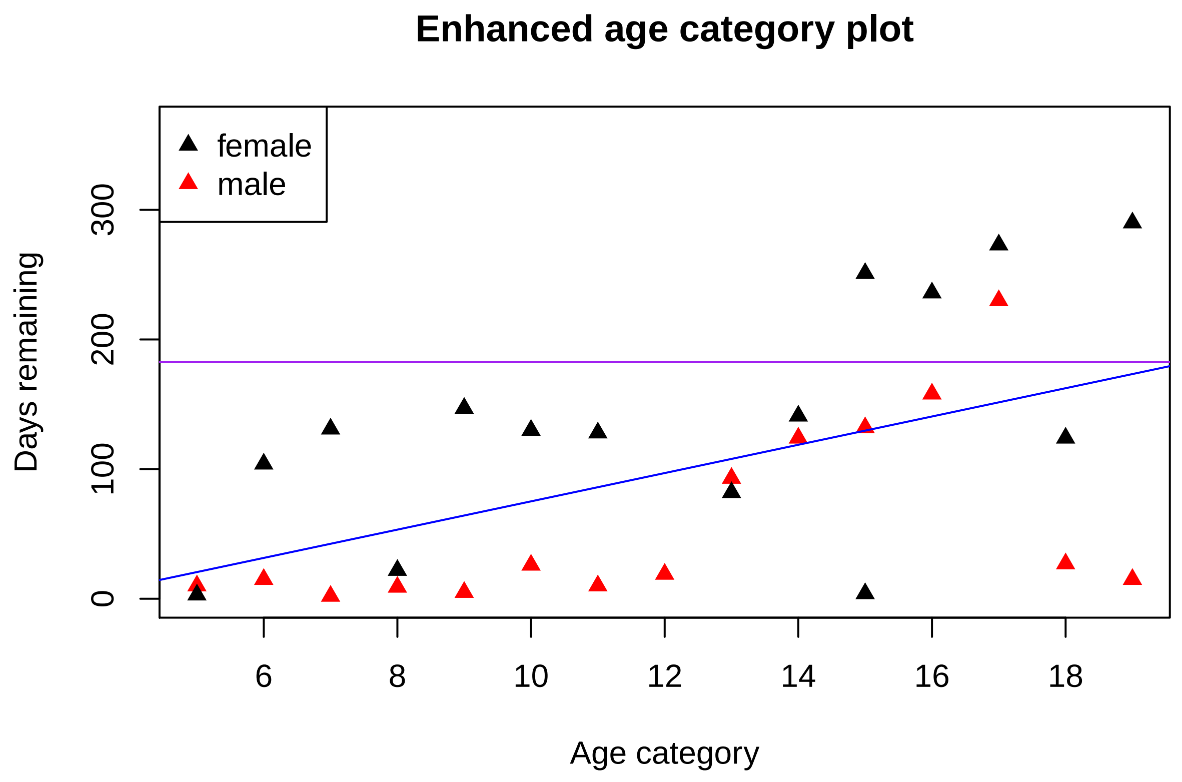


Surprisingly there's one outlier: Payton Payne's record set at age 11 improved all previous times run by girls age 12. A closer look shows that Payton holds records in five categories.

```
##   Age_category      Athlete         Date               Age
## 1            7 Payton Payne 25 July 2015  7 years, 234 days
## 2            9 Payton Payne  9 July 2017  9 years, 218 days
## 3           10 Payton Payne 26 July 2018 10 years, 235 days
## 4           11 Payton Payne 28 July 2019 11 years, 237 days
## 5           12 Payton Payne 28 July 2019 11 years, 237 days
```

Payton set her last record 28 July 2019. I assume in 2020 there were close to no competitions due to covid. I think it's fair to assume she would have further improved the age category 12 record had she gotten the chance. Therefore I decided to remove the outlier and to not count her outstanding performance in the age 12 categories.

Additionally I plot a half year horizontal line in purple and a least squares linear regression line in blue:



The linear regression (blue) shows nicely what one expects: While age has a big impact on the performance of young athletes, the impact is lost over time. It seems that young athletes improve their performance in close relation to getting older, while from maybe 16 onward the age factor appears to lose it's weight.

Age and adult athletes best performance

This prompted the follow up question: At what age did the all-time fastest athletes run their record times?

To answer this question I load the tables from the Wikipedia page sections 'All-time top 25' for men and women. I make sure to remove duplicate entries of record holders who ran their record time more than once.

To my disappointment, while the date when the record was run is present in the table, the age of the athlete's is missing. I decide to collect the links to the athlete's personal Wikipedia page, to then extract the age information from there. I use the following functions to first collect all links. This includes links to the cities where they competed and references. The second function then matches the athletes with their link based on their first names.

```
# all href in table with tableNr
hrefs_in_table = function(tableNr){
  links = url %>%
    read_html() %>%
    html_nodes('table.wikitable') %>%
    .[[tableNr]] %>%
    html_nodes('a') %>%
    html_attr('href') %>%
    paste0('https://en.wikipedia.org', .)
  (unique(links)) # remove duplicate entries
}

# add links column and match links to athlete
add_athlete_href = function(data_table, links){
  data_table$Link = NA
  for (idx in (1:length(data_table$Athlete))){
    athlete = data_table$Athlete[idx]
    name = strsplit(toString(athlete), " ")[[1]][1]
    matching_link = links[grep(name, links)]
    if(identical(matching_link, character(0))){next}
    data_table$Link[idx] = links[grep(name, links)]
  }
  return(data_table)
}
```

Sha'Carri Richardson's wiki link does not match the pattern, I add it explicitly. The tables format now looks like this:

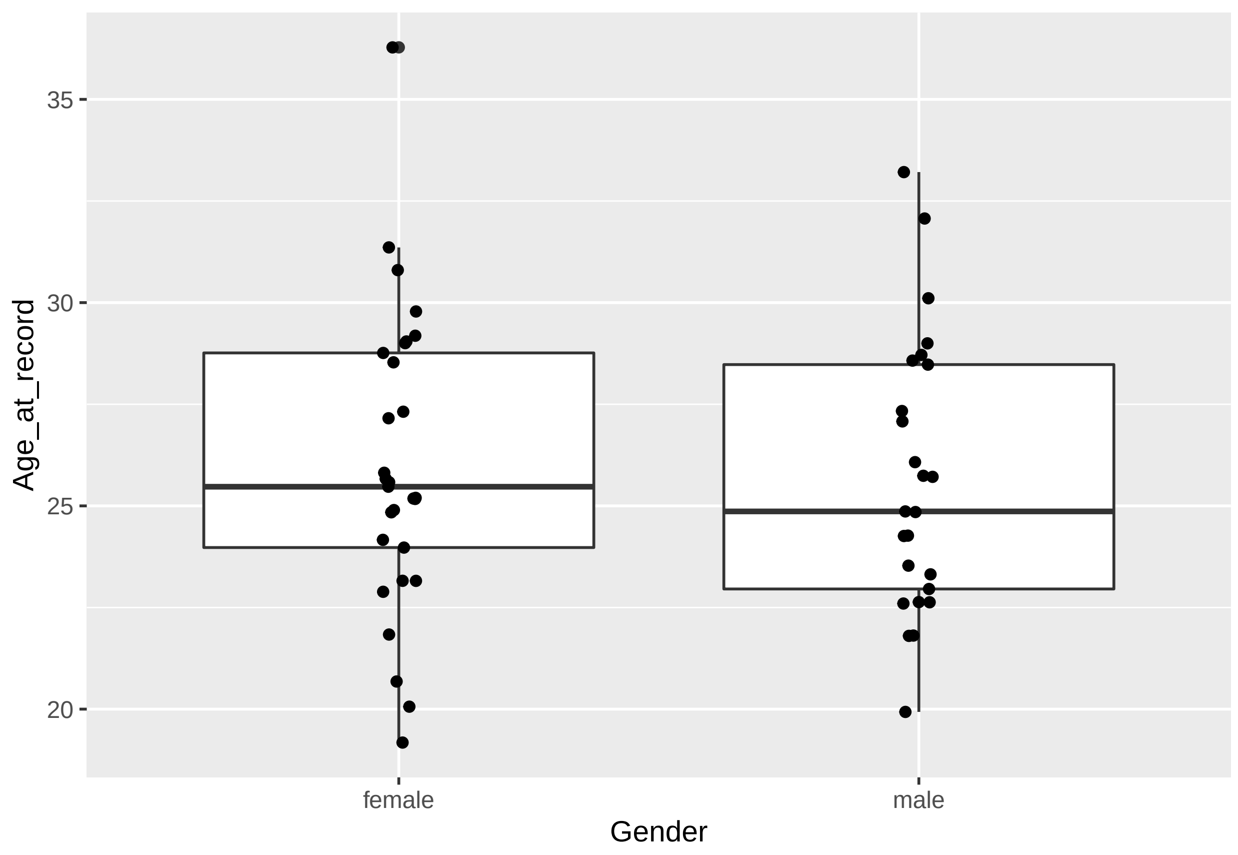
```
## Observations: 25
## Variables: 3
## $ Athlete <fct> Usain Bolt, Tyson Gay, Yohan Blake, Asafa Powell, Just...
## $ Date    <fct> 16 August 2009, 20 September 2009, 23 August 2012, 2 S...
## $ Link    <chr> "https://en.wikipedia.org/wiki/Usain_Bolt", "https://e...
```

Now lets collect the birthdays from their personal page. I take advantage of the typical info-box that's located on the top right of most wiki pages. However six of the entries aren't caught in the pattern (for various reasons) - I insert them in a post processing step.

```
# web scrawl for birthdays to be able to compute age at record
add_athlete_birthday = function(data_table){
  data_table$Born = NA
  for (idx in (1:length(data_table$Athlete))){
    pg = read_html(data_table$Link[idx])
    info_box <- html_nodes(pg, xpath="//table[@class = 'infobox vcard']")
    bday_node <- html_nodes(info_box, xpath = "//span[@class = 'bday']")
    bday = html_text(bday_node)[1]
    data_table$Born[idx] = bday
  }
  return(data_table)
}
```

Next I convert the dates when the records where run to standard date format "YYYY-MM-DD". Then I can calculate the age of the athletes at the time they ran their best race.

Bringing the two tables together we can analyse the data. The overall mean age at peak performance of the top 25 women and men amounts to 25.8 years. The athletes gender appears to have no significant influence on the age the athletes set their records.



The box plots show the data split in four quantiles (25% of the data per quantile): The middle line in the box is represented by the median, the box contains the upper and lower quantile, or the 50% of the data closest to the median. The whiskers visualize the remaining two, most extreme quantile groups. The age distribution looks quite similar between the genders. The respectable means are at 26.0 for female runners and 25.7 for males. The standard deviation for the female runners is 3.7 and for the males 3.4. My take is that I can not read a significant difference in prime age for athletic performances per genders from that data.

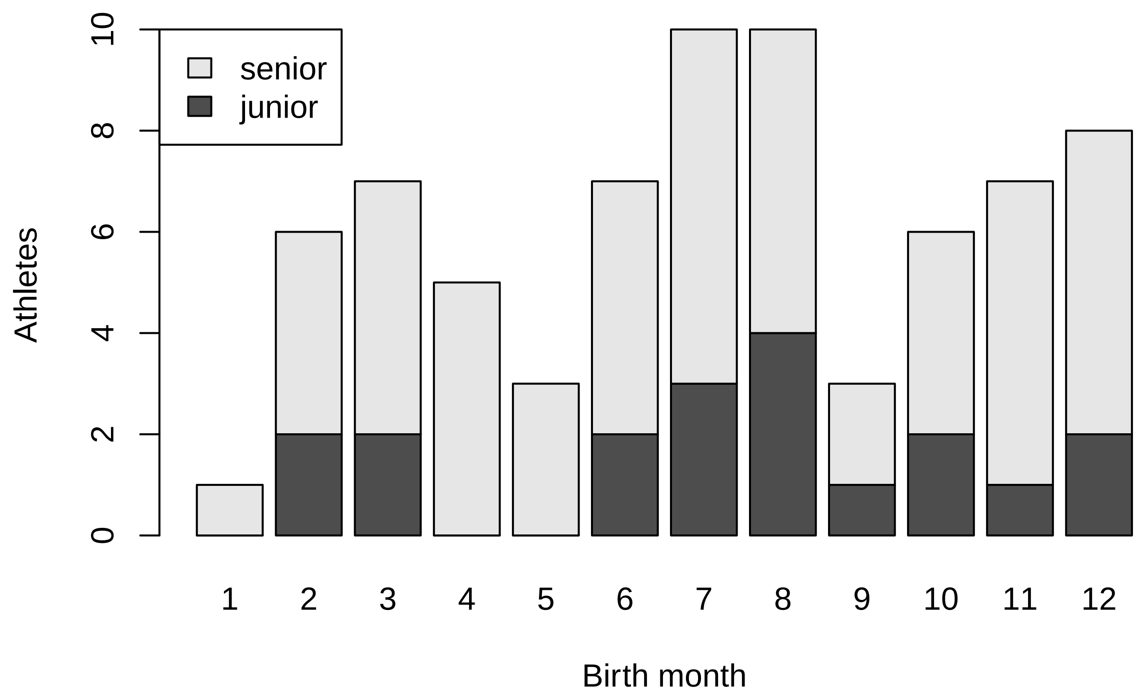
Best month to be born a sprinter

This brings me to the final question I want to address: Is there a best time of the year to be born an 100 meters athlete? This question is based on a hypothesis I heard before. Namely, that kids born at the end of the year would be advantaged in sports, as they are always the oldest in their age groups. This of course assumes that the categories are drawn by the birth year and not current age. If the categories are given by the current age, as suggested by the '100 meters per age category' section, then maybe one could still find a trend if there are fixed seasons of competitions. For example such a season could be during months with mild weather for outdoors competition.

To analyze this I gather the birth months for junior (per age category record holders) and senior (all time top 25) in a table. The month values of senior athletes can be directly read from the birthdays. From the juniors I calculate the birth month based on their record date and age at record.

```
## Observations: 73
## Variables: 3
## $ Athlete  <chr> "Kai Sapp", "Willie Washington", "Nyckoles Harbor", "...
## $ Month    <dbl> 6, 8, 7, 8, 10, 8, 9, 12, 7, 7, 10, 12, 6, 8, 3, 2, 2...
## $ Category <chr> "junior", "junior", "junior", "junior", "junior", "ju...
```

Lets illustrate the data in a bar plot.



For me it's hard to read any trend out of it by naked eye. It could follow a uniform distribution. This would indicate that the birth month has no influence on the chance to become a record setting athlete. To compare it with an uniform distribution lets compare the mean and variance to the expected mean and variance. We have a = 1, b = 12. The theoretical mean corresponds to (a+b)/2 = 6.5, the variance = ((b-a)^2)/b = 10.08333. The values we measure from the data seems to confirm a uniform distribution: mean is measured at 7.1 and variance 10.4. Given our data set of just 73 observations I believe this is comparable. I ran a maximum likelihood estimation algorithm to compare the generated estimators to a and b.

```
# install.packages("ExtDist")
library(ExtDist)
len = length(just_month$Month)
mle_uniform = eUniform(X = just_month$Month, method = "unbiased.MLE")
print(mle_uniform)
```

```
##
## Parameters for the Uniform distribution.
## (found using the  numerical.MLE method.)
##
##  Parameter     Type   Estimate
##          a boundary  0.9999878
##          b boundary 12.0000122
```

From what I understand this estimated the most likely parameters a and b given the data and assuming an uniform distribution. With that I conclude that the birth months in record setting 100 meters athletes appear to follow the uniform distribution. Hence if there's a best month to be born a runner, it did not manifest in this data.