Assignment 3

Feiyu Zheng (fz114)

2022/2/8

(1) Use the Master data frame in the Lahman package to create a tibble with exactly the same variables as the *babynames* data frame (except the *sex*), and ordered in the same way. For year, use the year of birth. For name, use the first name (variable *nameFirst*). The final table should look like this (where *prop* is the proportion of names in a specific birth year)

birthYear	nameFirst	n	prop

```r
# install.packages("Lahman") # install the packages
library(Lahman) # import the packages

master_t <- Master %>%
  as_tibble() %>%
  select(birthYear, nameFirst) %>%
  group_by(birthYear, nameFirst) %>%
  summarize(n = n()) %>% # compute the number of people with same first name
  mutate(prop = n / sum(n)) %>% # compute proportion
  ungroup() %>%
  arrange(birthYear, desc(n)) # sort the result first by birthYear then by n in descending order

master_t # show the result
```

```
## # A tibble: 12,996 x 4
##    birthYear nameFirst     n  prop
##        <int> <chr>     <int> <dbl>
##  1      1820 Alexander     1 1
##  2      1824 Henry         1 1
##  3      1832 Nate          1 0.5
##  4      1832 William       1 0.5
##  5      1835 Harry         1 1
##  6      1836 Dickey        1 1
##  7      1837 Morgan        1 1
##  8      1838 Bill          1 0.333
##  9      1838 Dave          1 0.333
## 10      1838 Lew           1 0.333
## # ... with 12,986 more rows
```

(2) Create a subset of the tibble created in (1) with first names that start with the letter "Y".

```
nameStartedWithY <- master_t %>%
  filter(str_starts(nameFirst, "Y")) # filter rows which first name starts with the letter Y

nameStartedWithY # show the result
```

```
## # A tibble: 66 x 4
##    birthYear nameFirst     n    prop
##        <int> <chr>     <int>   <dbl>
##  1      1859 Yank          1 0.0115
##  2      1869 Yale          1 0.00971
##  3      1873 Youngy        1 0.0116
##  4      1886 Yip           1 0.00719
##  5      1892 Yam           1 0.00662
##  6      1903 Yats          1 0.00917
##  7      1911 Yank          1 0.0102
##  8      1925 Yogi          1 0.0106
##  9      1928 Yo-Yo         1 0.00909
## 10      1967 Yorkis        1 0.00529
## # ... with 56 more rows
```

(3) Create a subset of the tibble created in (1) with first names that contain at least three vowels.

```
nameContainsThreeVowels <- master_t %>%
  filter(str_count(nameFirst, "[aeiou]") >= 3) # filter rows which first name contains three vowels

nameContainsThreeVowels # show the result
```

```
## # A tibble: 1,546 x 4
##    birthYear nameFirst      n   prop
##        <int> <chr>      <int>  <dbl>
##  1      1820 Alexander      1 1
##  2      1832 William        1 0.5
##  3      1840 Charlie        1 0.143
##  4      1840 George         1 0.143
##  5      1840 Washington     1 0.143
##  6      1843 Charlie        1 0.125
##  7      1844 Charlie        1 0.0556
##  8      1844 Cherokee       1 0.0556
##  9      1844 George         1 0.0556
## 10      1845 Freeman        1 0.0556
## # ... with 1,536 more rows
```

(4) In the Master dataframe, let us check whether the variable *birthYear* is consistent with the year in *birthDate*. Use a function in the *lubridate* package to extract the year from the *birthDate*. Call this variable *birthYear2*. In how many cases does *birthYear* have an "NA" entry? In how many cases does *birthYear2* have "NA" entry? In how many cases do both have "NA" entries? If you ignore all the cases with at least one "NA" entry (either in the *birthYear* or *birthYear2* variable), do all remaining cases match?

Create *birthYear2* by extracting year from *birthDate*

```
masterWithBirthYear2 <- Master %>%
  as_tibble() %>%
  mutate(birthYear2 = year(birthDate)) # create birthYear2 variable which stores the extracted year fro

masterWithBirthYear2 # show the result
```

```
## # A tibble: 20,093 x 27
##    playerID  birthYear birthMonth birthDay birthCountry birthState birthCity
##    <chr>         <int>      <int>    <int> <chr>        <chr>      <chr>
##  1 aardsda01      1981         12       27 USA          CO         Denver
##  2 aaronha01      1934          2        5 USA          AL         Mobile
##  3 aaronto01      1939          8        5 USA          AL         Mobile
##  4 aasedo01       1954          9        8 USA          CA         Orange
##  5 abadan01       1972          8       25 USA          FL         Palm Beach
##  6 abadfe01       1985         12       17 D.R.         La Romana  La Romana
##  7 abadijo01      1850         11        4 USA          PA         Philadelphia
##  8 abbated01      1877          4       15 USA          PA         Latrobe
##  9 abbeybe01      1869         11       11 USA          VT         Essex
## 10 abbeych01      1866         10       14 USA          NE         Falls City
## # ... with 20,083 more rows, and 20 more variables: deathYear <int>,
## #   deathMonth <int>, deathDay <int>, deathCountry <chr>, deathState <chr>,
## #   deathCity <chr>, nameFirst <chr>, nameLast <chr>, nameGiven <chr>,
## #   weight <int>, height <int>, bats <fct>, throws <fct>, debut <chr>,
## #   finalGame <chr>, retroID <chr>, bbrefID <chr>, deathDate <date>,
## #   birthDate <date>, birthYear2 <dbl>
```

Count rows which *birthYear* is "NA"

```
birthYearNA <- masterWithBirthYear2 %>%
  filter(is.na(birthYear)) %>% # filter rows which birthYear is NA
  nrow() # count total rows

print(paste("There are ", birthYearNA, " cases which birthYear is NA.", sep = "")) # show the result
```

```
## [1] "There are 114 cases which birthYear is NA."
```

Count rows which *birthYear2* is "NA"

```
birthYear2NA <- masterWithBirthYear2 %>%
  filter(is.na(birthYear2)) %>% # filter rows which birthYear2 is NA
  nrow() # count total rows

print(paste("There are ", birthYear2NA, " cases which birthYear2 is NA.", sep = "")) # show the result
```

```
## [1] "There are 423 cases which birthYear2 is NA."
```

Count rows which *birthYear* and *birthYear2* are both "NA"

```
birthYearBothNA <- masterWithBirthYear2 %>%
  filter(is.na(birthYear) & is.na(birthYear2)) %>% # filter rows which birthYear and birthYear2 are bot
  nrow() # count total rows

print(paste("There are ", birthYearBothNA, " cases which birthYear and birthYear2 are both NA.", sep =
```

```
## [1] "There are 114 cases which birthYear and birthYear2 are both NA."
```

Ignore all the cases with at least one "NA" entry and indicate how many cases have matched *birthYear* and *birthYear2*

```
casesWithoutNA <- masterWithBirthYear2 %>%
  filter(!is.na(birthYear) & !is.na(birthYear2)) %>% # filter rows which birthYear and birthYear2 are b
  nrow() # count total rows

print(paste("There are ", casesWithoutNA, " cases which birthYear and birthYear2 are both not NA.", sep
```

```
## [1] "There are 19670 cases which birthYear and birthYear2 are both not NA."
```

```
matchedCases <- masterWithBirthYear2 %>%
  filter(!is.na(birthYear) & !is.na(birthYear2) & birthYear == birthYear2) %>% # filter rows which birt
  nrow() # count total rows

print(paste("There are ", casesWithoutNA, " cases which birthYear and birthYear2 are both not NA and ar
```

```
## [1] "There are 19670 cases which birthYear and birthYear2 are both not NA and are equal to each othe
```

```
if(casesWithoutNA == matchedCases){
  print("All the remaining cases match.")
} else{
  print("There exists some unmatched cases in the remainin cases.")
}
```

```
## [1] "All the remaining cases match."
```

(5) Create a data frame of players showing just the playerID, first name, last name, given name, and career total (meaning, summed over all the years and all stints) of games (that is, the G variable) according to the Fielding data frame. [Hint: Join the Fielding data frame with the Master data frame]

```
playerCareer <- inner_join(Master, Fielding, by = "playerID") %>%
  group_by(playerID) %>%
  summarize(playerID, nameFirst, nameLast, nameGiven, careerTotal = sum(G)) %>% # compute the career to
  ungroup() %>%
  distinct() # clear redundant rows

playerCareer # show the result
```

```
## # A tibble: 19,698 x 5
##    playerID  nameFirst nameLast    nameGiven        careerTotal
##    <chr>     <chr>     <chr>       <chr>                  <int>
##  1 aardsda01 David     Aardsma     David Allan               331
##  2 aaronha01 Hank      Aaron       Henry Louis              3020
##  3 aaronto01 Tommie    Aaron       Tommie Lee                387
##  4 aasedo01  Don       Aase        Donald William            448
##  5 abadan01  Andy      Abad        Fausto Andres               9
##  6 abadfe01  Fernando  Abad        Fernando Antonio          384
##  7 abadijo01 John      Abadie      John W.                    12
##  8 abbated01 Ed        Abbaticchio Edward James              830
##  9 abbeybe01 Bert      Abbey       Bert Wood                  79
## 10 abbeych01 Charlie   Abbey       Charles S.                452
## # ... with 19,688 more rows
```

(6) Add a variable to your data frame obtained in (3) for full name by combining the first name and last name with a space between them.

```
Master %>%
  as_tibble() %>%
  mutate(fullName = paste(nameFirst, nameLast, sep = " ")) %>% # add the variable fullName by combining
  select(birthYear, nameFirst, fullName) %>%
  arrange(birthYear, nameFirst) %>%
  inner_join(nameContainsThreeVowels, by = c("birthYear" = "birthYear", "nameFirst" = "nameFirst")) # i
```

```
## # A tibble: 2,001 x 5
##    birthYear nameFirst  fullName                 n   prop
##        <int> <chr>      <chr>                <int>  <dbl>
##  1      1820 Alexander  Alexander Cartwright     1 1
##  2      1832 William    William Hulbert          1 0.5
##  3      1840 Charlie    Charlie Smith            1 0.143
##  4      1840 George     George Popplein          1 0.143
##  5      1840 Washington Washington Fulmer        1 0.143
##  6      1843 Charlie    Charlie Byrne            1 0.125
##  7      1844 Charlie    Charlie Mills            1 0.0556
##  8      1844 Cherokee   Cherokee Fisher          1 0.0556
##  9      1844 George     George Zettlein          1 0.0556
## 10      1845 Freeman    Freeman Brown            1 0.0556
## # ... with 1,991 more rows
```

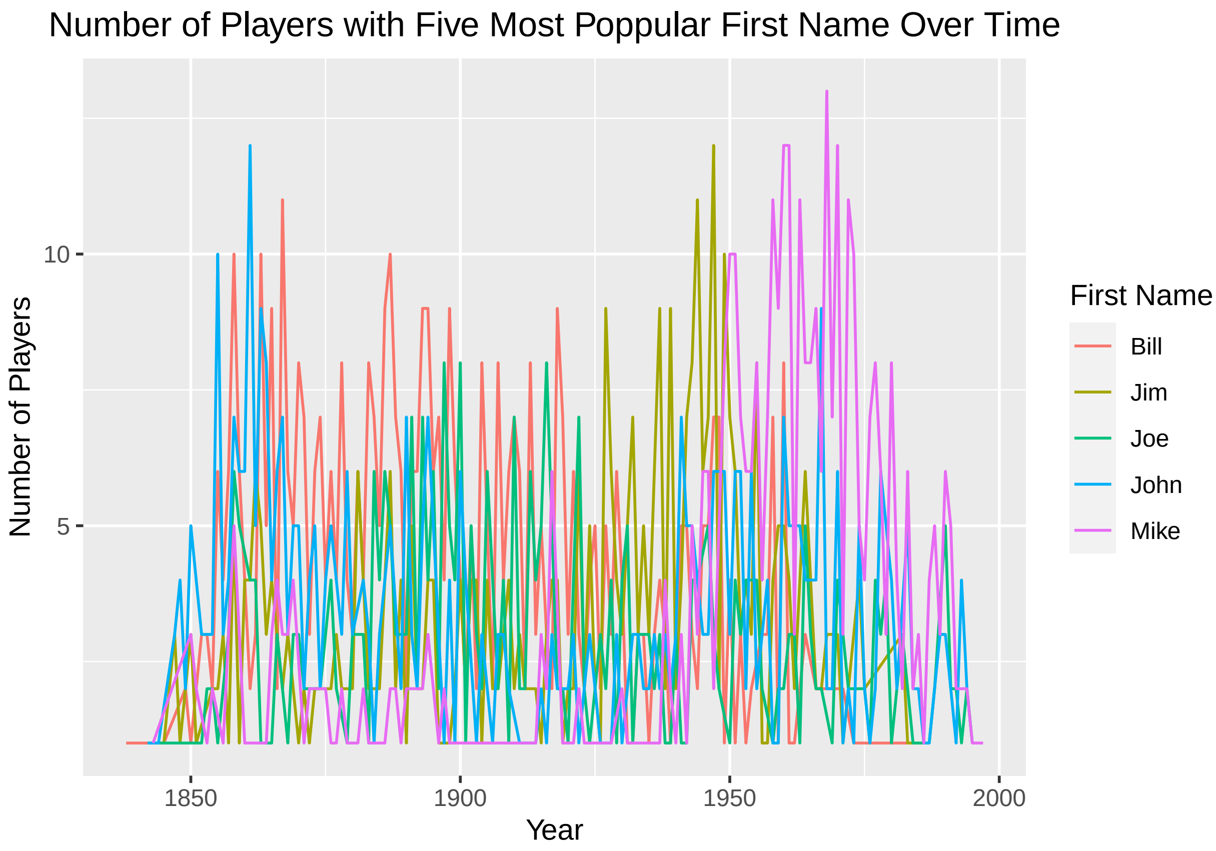
(7) Using the data frames you have created, or starting from scratch, determine the five most popular first names in baseball among players who played at least 500 games. Plot the number of players with these five most popular first names over time (according to the birth year) with lines in a single plot. Be sure to make the plot look nice by using a title and changing the axis labels if necessary.

```
top5PopularFirstName <- playerCareer %>%
  filter(careerTotal >= 500) %>% # filter rows which careerTotal is larger than 500
  group_by(nameFirst) %>%
  summarize(n = n()) %>% # count first name
  arrange(desc(n)) %>% # sort by n in descending order
  head(5) # get top 5 first names

top5PopularFirstName # show the result
```

```
## # A tibble: 5 x 2
##   nameFirst     n
##   <chr>     <int>
## 1 Mike         80
## 2 Joe          60
## 3 John         56
## 4 Bill         55
## 5 Jim          52
```

```
inner_join(Master, top5PopularFirstName, by = "nameFirst") %>% # use inner join to get rows which has t
  group_by(nameFirst, birthYear) %>%
  summarize(n = n()) %>% # compute number of players over year
  na.omit(birthYear) %>%
  ggplot(aes(x = birthYear, y = n, color = nameFirst)) +
  geom_line() + # line plot
  labs(
    x = "Year",
    y = "Number of Players",
    color = "First Name",
    title = "Number of Players with Five Most Poppular First Name Over Time"
  ) +
  theme(plot.title = element_text(hjust = 0.5))
```


Number of Players with Five Most Poppular First Name Over Time
Number of Players
Year
First Name
Bill
Jim
Joe
John
Mike