Assignment3

Rabin Paudel

9/24/2021

Problem 1

We can instill this library and load the dataset using the

```
library("dslabs")
#install.packages("dplyr")
library("dplyr")
```

folowing code chunk.

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library("tidyr")
admissions
```

```
##    major gender admitted applicants
## 1      A    men       62        825
## 2      B    men       63        560
## 3      C    men       37        325
## 4      D    men       33        417
## 5      E    men       28        191
## 6      F    men        6        373
## 7      A  women       82        108
## 8      B  women       68         25
## 9      C  women       34        593
## 10     D  women       35        375
## 11     E  women       24        393
## 12     F  women        7        341
```

Next use unite (also from the tidyr library) to combine the columns for

```
admissions_new <- admissions %>%
  pivot_longer(cols = c("admitted","applicants"),
                                        names_to = "name", values_to = "value")
admissions_new
```

name and gender into one column

```
## # A tibble: 24 x 4
##    major gender name       value
##    <chr> <chr>  <chr>      <dbl>
##  1 A     men    admitted      62
##  2 A     men    applicants   825
##  3 B     men    admitted      63
##  4 B     men    applicants   560
##  5 C     men    admitted      37
##  6 C     men    applicants   325
##  7 D     men    admitted      33
##  8 D     men    applicants   417
##  9 E     men    admitted      28
## 10 E     men    applicants   191
## # ... with 14 more rows
```

Finally use pivot_wider from the tidyr library to get the required

```
New_value <- admissions_new %>%
  unite(rate,name,gender)
New_value %>%
  pivot_wider(names_from = "rate")
```

data frame.

```
## # A tibble: 6 x 5
##   major admitted_men applicants_men admitted_women applicants_women
##   <chr>        <dbl>          <dbl>          <dbl>            <dbl>
## 1 A               62            825             82              108
## 2 B               63            560             68               25
## 3 C               37            325             34              593
## 4 D               33            417             35              375
## 5 E               28            191             24              393
## 6 F                6            373              7              341
```

Problem 2

This problem is from Chapter 5 of the book Modern Data Science with R.

The problem uses the Batting, Pitching, and Master data frames in the

```
#install.packages("Lahman")
library(Lahman)
#Batting
#Pitching
#Master
```

Lahman package.

First Part

Using the above data frames, answer the following questions.

Name every player in baseball history who has accumulated at least 300 home

runs (HR column) and at least 300 stolen bases (SB column). You can find

the first and last name of the player in the Master data frame. Join this

to your result along with the total home runs and total bases stolen for

```
mas <- Master %>%
  group_by(playerID) %>%
  summarize( playerID = paste(playerID),
             name = paste(nameFirst, nameLast, sep = " "))

fg <- Batting %>%
  group_by( playerID) %>%
  summarize(HR = sum(HR), SB = sum(SB))
fg %>% left_join(mas, by = "playerID") %>%
  filter(HR >= 300 & SB >= 300)%>%
  arrange(HR)
```

each of these elite players.

```
## # A tibble: 8 x 4
##   playerID     HR    SB name
##   <chr>     <int> <int> <chr>
## 1 finlest01   304   320 Steve Finley
## 2 sandere02   305   304 Reggie Sanders
## 3 bondsbo01   332   461 Bobby Bonds
## 4 beltrca01   435   312 Carlos Beltran
## 5 dawsoan01   438   314 Andre Dawson
## 6 mayswi01    660   338 Willie Mays
## 7 rodrial01   696   329 Alex Rodriguez
## 8 bondsba01   762   514 Barry Bonds
```

```
Batting %>%
  group_by(playerID) %>%
  summarise(HR = sum(HR), SB = sum(SB)) %>%
  filter(HR >= 300 & SB >= 300) %>%
  left_join(Master, by = "playerID") %>%
  select(nameFirst, nameLast, HR, SB)
```

Or I like to combine above data and get same result in same codu chunks.

```
## # A tibble: 8 x 4
##   nameFirst nameLast     HR    SB
##   <chr>     <chr>     <int> <int>
## 1 Carlos    Beltran     435   312
## 2 Barry     Bonds       762   514
## 3 Bobby     Bonds       332   461
## 4 Andre     Dawson      438   314
## 5 Steve     Finley      304   320
## 6 Willie    Mays        660   338
## 7 Alex      Rodriguez   696   329
## 8 Reggie    Sanders     305   304
```

Second Part

Similarly, name every pitcher in baseball history who has accumulated at

```
Pitching %>%
  group_by(playerID) %>%
  summarize(W = sum(W), SO = sum(SO)) %>%
  filter(W >= 300 & SO >= 3000) %>%
  left_join(mas, by = "playerID") %>%
  arrange(W)
```

least 300 wins (W column) and at least 3,000 strikeouts (SO column).

```
## # A tibble: 10 x 4
##    playerID      W    SO name
##    <chr>     <int> <int> <chr>
##  1 johnsra05   303  4875 Randy Johnson
##  2 seaveto01   311  3640 Tom Seaver
##  3 perryga01   314  3534 Gaylord Perry
##  4 niekrph01   318  3342 Phil Niekro
##  5 ryanno01    324  5714 Nolan Ryan
##  6 suttodo01   324  3574 Don Sutton
##  7 carltst01   329  4136 Steve Carlton
##  8 clemero02   354  4672 Roger Clemens
##  9 maddugr01   355  3371 Greg Maddux
## 10 johnswa01   417  3509 Walter Johnson
```

```
Pitching %>%
  group_by(playerID) %>%
  summarise(W = sum(W), SO = sum(SO)) %>%
  filter(W >= 300 & SO >= 3000) %>%
  left_join(Master, by = "playerID") %>%
  select(nameFirst, nameLast, W, SO)
```

Or other way

```
## # A tibble: 10 x 4
##    nameFirst nameLast     W    SO
##    <chr>     <chr>    <int> <int>
##  1 Steve     Carlton    329  4136
##  2 Roger     Clemens    354  4672
##  3 Randy     Johnson    303  4875
##  4 Walter    Johnson    417  3509
##  5 Greg      Maddux     355  3371
##  6 Phil      Niekro     318  3342
##  7 Gaylord   Perry      314  3534
##  8 Nolan     Ryan       324  5714
##  9 Tom       Seaver     311  3640
## 10 Don       Sutton     324  3574
```

Third Part

Identify the name and year of every player who has hit at least 50 home runs in a single season. Let table1 refer to the data frame that contains this information. For each season that appeared in the data frame table1, find the player that has the lowest batting average that season.

```
mas <- Master %>%
  group_by(playerID) %>%
  summarize( playerID = paste(playerID),
             name = paste(nameFirst, nameLast, sep = " "))
fg <- Batting %>%
  group_by( playerID, yearID) %>%
  select(playerID, yearID, HR,H,AB)
fg1 <- fg %>%
  semi_join(mas, by = "playerID") %>%
  filter(HR >= 50)
```

Hint: Use a semi_join

first dentify the name and year of every player who has hit at least 50

```
fg1 %>% left_join(mas, by = "playerID") %>%
  arrange(desc(HR))
```

home runs in a single season

```
## # A tibble: 45 x 6
## # Groups:   playerID, yearID [45]
##    playerID  yearID    HR     H    AB name
##    <chr>      <int> <int> <int> <int> <chr>
##  1 bondsba01   2001    73   156   476 Barry Bonds
##  2 mcgwima01   1998    70   152   509 Mark McGwire
##  3 sosasa01    1998    66   198   643 Sammy Sosa
##  4 mcgwima01   1999    65   145   521 Mark McGwire
##  5 sosasa01    2001    64   189   577 Sammy Sosa
##  6 sosasa01    1999    63   180   625 Sammy Sosa
##  7 marisro01   1961    61   159   590 Roger Maris
##  8 ruthba01    1927    60   192   540 Babe Ruth
##  9 ruthba01    1921    59   204   540 Babe Ruth
## 10 stantmi03   2017    59   168   597 Giancarlo Stanton
## # ... with 35 more rows
```

```
last <- Batting %>%
  group_by(playerID, yearID) %>%
  filter(HR >= 50) %>%
  mutate(average = sum(H)/sum(AB)) %>%
  select(playerID, yearID, HR, average) %>%
  arrange(average)

mas %>%
  right_join(last, by = "playerID") %>%
  select(HR,average,name,yearID) %>%
  arrange(average)%>%
  ungroup()
```

find the player that has the lowest batting average that season

```
## # A tibble: 45 x 4
##       HR average name              yearID
##    <int>   <dbl> <chr>              <int>
##  1    53   0.260 Pete Alonso         2019
##  2    54   0.260 Jose Bautista       2010
##  3    51   0.263 Andruw Jones        2005
##  4    61   0.269 Roger Maris         1961
##  5    50   0.272 Greg Vaughn         1998
##  6    51   0.277 Cecil Fielder       1990
```

```
##  7    65   0.278 Mark McGwire        1999
##  8    59   0.281 Giancarlo Stanton   2017
##  9    52   0.284 Aaron Judge         2017
## 10    56   0.284 Ken Griffey         1998
## # ... with 35 more rows
```

```
Batting %>%
  group_by(playerID, yearID) %>%
  summarise(HR = sum(HR), average = sum(H)/sum(AB)) %>%
  filter(HR >= 50) %>%
  left_join(Master, by = "playerID") %>%
  select(nameFirst, nameLast, HR, average) %>%
  ungroup() %>%
  arrange(average)
```

Or Other way we get same result.

```
## `summarise()` has grouped output by 'playerID'. You can override using the `.groups` argument.
```

```
## Adding missing grouping variables: `playerID`
```

```
## # A tibble: 46 x 5
##    playerID  nameFirst nameLast    HR average
##    <chr>     <chr>     <chr>    <int>   <dbl>
##  1 alonspe01 Pete      Alonso      53   0.260
##  2 bautijo02 Jose      Bautista    54   0.260
##  3 jonesan01 Andruw    Jones       51   0.263
##  4 marisro01 Roger     Maris       61   0.269
##  5 vaughgr01 Greg      Vaughn      50   0.272
##  6 mcgwima01 Mark      McGwire     58   0.274
##  7 fieldce01 Cecil     Fielder     51   0.277
##  8 mcgwima01 Mark      McGwire     65   0.278
##  9 stantmi03 Giancarlo Stanton     59   0.281
## 10 judgeaa01 Aaron     Judge       52   0.284
## # ... with 36 more rows
```

THE END