DSCI 401 HW 3

Isabel Heard

10/06/2023

Contents

1

Use the Batting, Pitching, and People tables in the Lahman package to answer the following questions:

```
#Install libraries
library(Lahman)
library("dplyr")
```

```
##
## 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")
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

```
#Load in Lahman tables
data(Batting)
head(Batting)
```

```
data(Pitching)
head(Pitching)

data(People)
head(People)
```

A

Name every player in baseball history who has accumulated at least 300 home runs (HR) AND at least 300 stolen bases (SB). You can find the first and last name of the player in the People data frame. Join this to your result along with the total home runs and total bases stolen for each of these elite players.

```
#Name (People), HR (Batting), SB(Batting)
#Common variable = playerID
Batting %>%
  group_by(playerID) %>%
  summarize(total_HR = sum(HR),total_SB = sum(SB)) %>%
  left_join(People, by = c("playerID" = "playerID")) %>%
  filter(total_HR >= 300 & total_SB >= 300) %>%
  select(nameGiven, total_HR, total_SB)
```

```
## # A tibble: 8 x 3
##   nameGiven          total_HR total_SB
##   <chr>                 <int>    <int>
## 1 Carlos Ivan             435      312
## 2 Barry Lamar             762      514
## 3 Bobby Lee               332      461
## 4 Andre Nolan             438      314
## 5 Steven Allen            304      320
## 6 Willie Howard           660      338
## 7 Alexander Enmanuel      696      329
## 8 Reginald Laverne        305      304
```

B

Similarly, name every pitcher in baseball history who has accumulated at least 300 wins (W) and at least 3,000 strikeouts (SO).

```
#Name (People), W (Pitching), SO (Pitching)
#Common variable = playerID
Pitching %>%
  group_by(playerID) %>%
  summarize(TotalWin = sum(W),TotalSO = sum(SO)) %>%
  left_join(People, by = c("playerID" = "playerID")) %>%
  filter(TotalWin >= 300 & TotalSO >= 3000) %>%
  select(nameFirst, nameLast, nameGiven, TotalWin, TotalSO)
```

```
## # A tibble: 10 x 5
##    nameFirst nameLast nameGiven       TotalWin TotalSO
##    <chr>     <chr>    <chr>              <int>   <int>
##  1 Steve     Carlton  Steven Norman        329    4136
##  2 Roger     Clemens  William Roger        354    4672
##  3 Randy     Johnson  Randall David        303    4875
##  4 Walter    Johnson  Walter Perry         417    3509
##  5 Greg      Maddux   Gregory Alan         355    3371
```

```
##  6 Phil      Niekro   Philip Henry         318    3342
##  7 Gaylord   Perry    Gaylord Jackson      314    3534
##  8 Nolan     Ryan     Lynn Nolan           324    5714
##  9 Tom       Seaver   George Thomas        311    3640
## 10 Don       Sutton   Donald Howard        324    3574
```

C

Identify the name and year of every player who has hit at least 50 home runs in a single season. Which player had the lowest batting average in that season? (Note: batting average)

```
#Name (People), HR (Batting), AB (Batting), H (Batting), yearID (Batting)
#Common variable = playerID
Batting %>%
  group_by(playerID, yearID) %>%
  summarize(TotalHR = sum(HR), BA = sum(H)/sum(AB)) %>%
  right_join(People, by = c("playerID" = "playerID")) %>%
  filter(TotalHR >= 50) %>%
  select(nameFirst, nameLast, yearID, TotalHR, BA) %>%
  arrange(BA)
```

```
## `summarise()` has grouped output by 'playerID'. You can override using the
## `.groups` argument.
## Adding missing grouping variables: `playerID`
```

```
## # A tibble: 47 x 6
## # Groups:   playerID [30]
##    playerID  nameFirst nameLast yearID TotalHR    BA
##    <chr>     <chr>     <chr>     <int>   <int> <dbl>
##  1 alonspe01 Pete      Alonso     2019      53 0.260
##  2 bautijo02 Jose      Bautista   2010      54 0.260
##  3 jonesan01 Andruw    Jones      2005      51 0.263
##  4 marisro01 Roger     Maris      1961      61 0.269
##  5 vaughgr01 Greg      Vaughn     1998      50 0.272
##  6 mcgwima01 Mark      McGwire    1997      58 0.274
##  7 fieldce01 Cecil     Fielder    1990      51 0.277
##  8 mcgwima01 Mark      McGwire    1999      65 0.278
##  9 stantmi03 Giancarlo Stanton    2017      59 0.281
## 10 judgeaa01 Aaron     Judge      2017      52 0.284
## # ... with 37 more rows
```

2

Use the nycflights13 package and the flights and planes tables to answer the following questions:

```
#Install libraries
#install.packages('nycflights13')
library(nycflights13)

#Load in nycflights13 tables
data(flights)
#head(flights)

data(planes)
#head(planes)
```

A

What is the oldest plane (specified by the tailnum variable) that flew from New York City airports in 2013?

```
#no need to finter for 2013
#common variable is tailnum
#use head to get first row in data frame
planes %>%
  rename(year_built = year) %>%
  left_join(flights, by = "tailnum") %>%
  arrange(year_built) %>%
  select(tailnum, year_built) %>%
  head(1)
```

```
## Warning in left_join(., flights, by = "tailnum"): Each row in `x` is expected to match at most 1 row
## i Row 1 of `x` matches multiple rows.
## i If multiple matches are expected, set `multiple = "all"` to silence this
##   warning.
```

```
## # A tibble: 1 x 2
##   tailnum year_built
##   <chr>        <int>
## 1 N381AA        1956
```

B

How many airplanes that flew from New York City are included in the planes table?

```
#find distinct planes
#common variable is tailnum
planes_nyc <- flights %>%
  inner_join(planes, by = "tailnum") %>%
  summarize(n=n_distinct(tailnum))
print(planes_nyc)
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1  3322
```

3

Generate the code to convert the following data frame to wide format.

```
dat <- data.frame(grp = c("A","A","B","B"),
                  sex = c("F","M","F","M"),
                  meanL = c(0.225,0.47,0.325,0.547),
                  sdL = c(0.106,.325,.106,.308),
                  meanR = c(.34,.57,.4,.647),
                  sdR = c(0.0849, 0.325, 0.0707, 0.274))
#print(dat)
```

```
wide_dat <- dat %>%
  pivot_wider(
    names_from = sex,
    values_from = c("meanL", "sdL", "meanR", "sdR"),
```

```
    names_sep = ".")
print(wide_dat)
```

```
## # A tibble: 2 x 9
##   grp   meanL.F meanL.M sdL.F sdL.M meanR.F meanR.M  sdR.F sdR.M
##   <chr>   <dbl>   <dbl> <dbl> <dbl>   <dbl>   <dbl>  <dbl> <dbl>
## 1 A       0.225   0.47  0.106 0.325    0.34   0.57  0.0849 0.325
## 2 B       0.325   0.547 0.106 0.308    0.4    0.647 0.0707 0.274
```

4

Consider the pccc_icd10 dataset.

```
df <- read.csv("https://raw.githubusercontent.com/gjm112/DSCI401/main/data/pccc_icd10_dataset.csv")
```

A

Remove all the columns labeled with "g" and a number.

```
# ^ is referencing the start of the string
# //d+ matches one or more digits
# - sign before matches removes those values from the data set
df_filtered <- df %>%
  select(-matches("^g\\d+"))
head(df_filtered)
```

```
##   id     dx1     dx2     dx3     dx4     dx5     dx6     dx7     dx8     dx9
## 1  1 S9410XS  I67841  E70339    <NA> S14121A  M66229 S92065G   O0973    <NA>
## 2  2    <NA> S53422D S92244B  M66342    <NA> S32442A T1582XD S72325C S52131B
## 3  3    <NA> S91225S    <NA> W6119XD   C8397 M80819K S72114R    <NA> Y382X3D
## 4  4 S7226XK   Y93G2   L0592  K08530    <NA> S62637D T84612A    <NA>    <NA>
## 5  5 S92246A   O4212   D2920 S42434S  F15980    <NA> S52572R M8080XA X731XXD
## 6  6    <NA> S52291C    <NA>    <NA>   E7140  H05222 S60549S    <NA> S32616G
##      dx10     pc1     pc2     pc3     pc4     pc5     pc6     pc7     pc8
## 1    <NA> 0PSH3CZ 0JPT3XZ 037906Z 0JHD3HZ 0KQ54ZZ 0WPK3YZ 01B04ZX 0DWV07Z
## 2   O1400 0DVM7DZ 0NRJ47Z DWY48ZZ 0HRWX7Z BP091ZZ 0Y0H4JZ    <NA> 0B9880Z
## 3  I70519 0PBV4ZX 0XM20ZZ 0DWD4UZ 2W07XYZ F0636ZZ 0RUP37Z    <NA> 0WCP8ZZ
## 4    <NA> DDY37ZZ 07LL0CZ 0Y9930Z 037M3GZ 04100Z4    <NA> 0SPG33Z 0TRC07Z
## 5 S42471K 02UL4KZ 03VD0ZZ 02110K8 3E050HZ 3E0U0GB    <NA> 0SPQ30Z 0WWBXYZ
## 6    <NA> 0D740DZ 0V1Q4JJ 10A07Z6 03150AK 047J47Z 0NQHXZZ 08BY3ZZ 047B376
##       pc9    pc10
## 1 09513ZZ 0V554ZZ
## 2    <NA>    <NA>
## 3 0DUM4KZ BN02ZZZ
## 4 041M0KQ DB10B8Z
## 5    <NA> 0SWN38Z
## 6 0SRQ07Z 0GPR00Z
```

B

Convert this to a long data set with three columns: id, type (pc or dx), and code.

```
# ^(dx|pc)\\d+$ match column names that start with dx or pc followed by one or more digit and populate
df_long <- df_filtered %>%
  pivot_longer(
```

```
    cols = -id,
    names_to = "type",
    names_pattern = "^(dx|pc)\\d+$"
  ) %>%
  filter(type %in% c("dx", "pc"))

df_long <- df_long %>%
  rename(code = value)
head(df_long)
```

```
## # A tibble: 6 x 3
##      id type  code
##   <int> <chr> <chr>
## 1     1 dx    S9410XS
## 2     1 dx    I67841
## 3     1 dx    E70339
## 4     1 dx    <NA>
## 5     1 dx    S14121A
## 6     1 dx    M66229
```

Colab Link

https://colab.research.google.com/drive/1wxj40CAssvcrznMYU8GT8vKXvyvpe7oU?usp=sharing