HW3

2023-10-03

```r
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.3     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.3     ✔ tibble    3.2.1
## ✔ lubridate 1.9.2     ✔ tidyr     1.3.0
## ✔ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflic
ts to become errors
```

```r
library(mdsr)
library(nycflights13)
library(Lahman)
```

QUESTION 1 Use the Batting, Pitching, and People tables in the Lahman package to answer the following questions:

a. Name every player in baseball history who has accumu- lated 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.

```r
#variable for total HR and variable for total SB by playerID
batter_summary <- Batting %>% group_by(playerID) %>%
  summarize(tHR=sum(HR),tSB=sum(SB)) %>%
  #Filter total HR and total SB.
  filter(tHR>=300, tSB>300)
#Join with 'people' to show players' names
batter_summary %>%
  left_join(People %>% select(playerID, nameFirst, nameLast), by = c("playerID" = "pla
yerID"))
```

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

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

```r
#variables to show total W and total SO per playerID
pitcher_summary <- Pitching %>% group_by(playerID) %>%
  summarize(wins=sum(W), strikeouts=sum(SO)) %>%
  #Filter total W and total SO.
  filter(wins>=300, strikeouts>=3000)
#Join with 'people' to show players' names.
pitcher_summary %>%
  left_join(People %>% select(playerID, nameFirst, nameLast), by = c("playerID" = "pla
yerID"))
```

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

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?

```r
#variables for total HR and batting average per player, per year.
Batting %>% group_by(playerID, yearID) %>%
  summarize(HR = sum(HR), bat_ave = sum(H)/sum(AB)) %>%
  #Filter total HR
  filter(HR>=50) %>%
  #Join with 'people' to get players' names.
  left_join(People %>% select(playerID, nameFirst, nameLast), by = c("playerID" = "pla
yerID")) %>%
  arrange(bat_ave) %>% head(1)
```

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

```
## # A tibble: 1 × 6
## # Groups:   playerID [1]
##   playerID  yearID    HR bat_ave nameFirst nameLast
##   <chr>      <int> <int>   <dbl> <chr>     <chr>
## 1 alonspe01   2019    53   0.260 Pete      Alonso
```

Pete Alonso

QUESTION 2 Use the nycflights13 package and the flights and planes tables to answer the following questions:

```r
head(planes)
```

```
## # A tibble: 6 × 9
##   tailnum  year type                  manufacturer model engines seats speed engine
##   <chr>   <int> <chr>                 <chr>        <chr>   <int> <int> <int> <chr>
## 1 N10156   2004 Fixed wing multi … EMBRAER      EMB-…       2    55    NA Turbo…
## 2 N102UW   1998 Fixed wing multi … AIRBUS INDU… A320…       2   182    NA Turbo…
## 3 N103US   1999 Fixed wing multi … AIRBUS INDU… A320…       2   182    NA Turbo…
## 4 N104UW   1999 Fixed wing multi … AIRBUS INDU… A320…       2   182    NA Turbo…
## 5 N10575   2002 Fixed wing multi … EMBRAER      EMB-…       2    55    NA Turbo…
## 6 N105UW   1999 Fixed wing multi … AIRBUS INDU… A320…       2   182    NA Turbo…
```

```r
head(flights)
```

```
## # A tibble: 6 × 19
##    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
## 1  2013     1     1      517            515         2      830            819
## 2  2013     1     1      533            529         4      850            830
## 3  2013     1     1      542            540         2      923            850
## 4  2013     1     1      544            545        -1     1004           1022
## 5  2013     1     1      554            600        -6      812            837
## 6  2013     1     1      554            558        -4      740            728
## # ℹ 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
## #   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dttm>
```

a. What is the oldest plane (specified by the tailnum vari- able) that flew from New York City airports in 2013?

```r
#select relevant columns
old_plane <- flights %>% select(tailnum, origin) %>%
  #join with selected 'planes' columns
  left_join(planes %>% select(tailnum, year), by = c("tailnum" = "tailnum")) %>%
  #Group by tailnum and year to ensure each tailnum only appears once in the tailnum c
olumn.
  group_by(tailnum,year) %>%
  summarise(count=n()) %>%
  #Arrange to get plane with earliest year.
  arrange(year)
```

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

```r
head(old_plane,1)
```

```
## # A tibble: 1 × 3
## # Groups:   tailnum [1]
##   tailnum  year count
##   <chr>   <int> <int>
## 1 N381AA   1956    22
```

N381AA is the oldest plane, created in 1956.

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

```r
#select relevant columns
incl_plane <- flights %>% select(tailnum, origin) %>%
  #group by tailnum to ensure each tailnum only appears once in the tailnum column.
  group_by(tailnum) %>%
  summarise(nyplane = n()) %>%
  #Join selected 'planes' columns and remove any years with null values.
  inner_join(planes %>% select(tailnum,year), by = c("tailnum"="tailnum")) %>%
  filter(!is.na(year)) %>%
  arrange("tailnum")

count(incl_plane)
```

```
## # A tibble: 1 × 1
##       n
##   <int>
## 1  3252
```

QUESTION 3 Convert the following data frame to wide format

```r
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)
                  )
#Use pivot_wider() to convert to a wider format.
dat %>% pivot_wider(names_from = sex, values_from = c(meanL, sdL, meanR,sdR))
```

```
## # A tibble: 2 × 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
```

QUESTION 4 Consider the pccc icd10 dataset.

```r
library(pccc)
head(pccc_icd10_dataset)
```

```
##   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     g1     g2     g3     g4     g5     g6     g7     g8
## 1 09513ZZ 0V554ZZ 239196 672832 683784 757546     NA 168052 104625     NA
## 2    <NA>    <NA> 931331 404900 912213     NA 964580 371556 778488 115827
## 3 0DUM4KZ BN02ZZZ 627455 638100 745829 843799 322975     NA     NA 932106
## 4 041M0KQ DB10B8Z 809782 153243 413723 130995 211708 610135     NA 471383
## 5    <NA> 0SWN38Z     NA 636794     NA 928572 930823 168586 133292 699936
## 6 0SRQ07Z 0GPR00Z 281891 318962 542326 705580 700647 929863 338026 525937
##       g9    g10
## 1 850974     NA
## 2 440619 955264
## 3 289004 242699
## 4 191245 135116
## 5 500743     NA
## 6 412691     NA
```

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

```r
#select columns with names don't start with g (^g) followed by a number ([0-9])
pccc_new <- select(pccc_icd10_dataset, -matches("^g[0-9]"))
head(pccc_new)
```

```
##   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 the data set from (a) to a long data set with three columns: id, type (pc or dx), and code.

```r
#pivot_longer() to convert to longer format.
pccc_new %>% pivot_longer(-id, names_to = "type", values_to = "code")
```

```
## # A tibble: 20,000 × 3
##       id type  code
##    <int> <chr> <chr>
##  1     1 dx1   S9410XS
##  2     1 dx2   I67841
##  3     1 dx3   E70339
##  4     1 dx4   <NA>
##  5     1 dx5   S14121A
##  6     1 dx6   M66229
##  7     1 dx7   S92065G
##  8     1 dx8   O0973
##  9     1 dx9   <NA>
## 10     1 dx10  <NA>
## # ℹ 19,990 more rows
```

Python Version: https://colab.research.google.com/drive/1IuNwaOdr2CMAaYrfnZqDl5xEkfEWWqZ2?usp=sharing