hw_submit_wk8

STAT 412/612 Week 8 Homework

Yunting

3/7/2020

Exercise 1: Keys

key = a set of variables used to connect a pair of dataframes

primary key = identifies individual rows its own data frame. Give me a value, I can tell the unique row that has that value.

1. Read the description of the babynames dataset with (you might need to install babynames)

```
library(tidyverse)
```

```
## -- Attaching packages --- tidyverse 1.3.0 --

## <U+2713> ggplot2 3.3.0     <U+2713> purrr   0.3.3
## <U+2713> tibble  2.1.3     <U+2713> dplyr   0.8.3
## <U+2713> tidyr   1.0.0     <U+2713> stringr 1.4.0
## <U+2713> readr   1.3.1     <U+2713> forcats 0.4.0

## -- Conflicts ------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(help = "babynames")
```

What are the data frames in this data set?

Ans: Applicants, Baby names, Births and Lifetables.

```
library(babynames)
```

```
## Warning: package 'babynames' was built under R version 3.6.3
```

```
head(applicants) #Applicants.
```

```
## # A tibble: 6 x 3
##    year sex    n_all
##   <int> <chr>  <int>
## 1  1880 F      97605
## 2  1880 M     118400
## 3  1881 F      98855
## 4  1881 M     108282
## 5  1882 F     115695
## 6  1882 M     122031
```

```
head(babynames) #Baby names.
```

```
## # A tibble: 6 x 5
##    year sex   name          n   prop
##   <dbl> <chr> <chr>     <int>  <dbl>
## 1  1880 F     Mary       7065 0.0724
## 2  1880 F     Anna       2604 0.0267
## 3  1880 F     Emma       2003 0.0205
## 4  1880 F     Elizabeth  1939 0.0199
## 5  1880 F     Minnie     1746 0.0179
## 6  1880 F     Margaret   1578 0.0162
```

```
head(births) # Births
```

```
## # A tibble: 6 x 2
##    year  births
##   <int>   <int>
## 1  1909 2718000
## 2  1910 2777000
## 3  1911 2809000
## 4  1912 2840000
## 5  1913 2869000
## 6  1914 2966000
```

```
head(lifetables) # Lifetables
```

```
## # A tibble: 6 x 9
##       x      qx     lx    dx    Lx      Tx    ex sex    year
##   <dbl>   <dbl>  <dbl> <dbl> <dbl>   <dbl> <dbl> <fct> <dbl>
## 1     0 0.146   100000 14596 90026 5151511  51.5 M      1900
## 2     1 0.0328   85404  2803 84003 5061484  59.3 M      1900
## 3     2 0.0163   82601  1350 81926 4977482  60.3 M      1900
## 4     3 0.0105   81251   855 80824 4895556  60.2 M      1900
## 5     4 0.00875  80397   703 80045 4814732  59.9 M      1900
## 6     5 0.00628  79693   501 79443 4734687  59.4 M      1900
```

What are the keys in each data frame? Demonstrate they are unique.

```
# applicants : the keys are year and sex.
applicants %>%
  group_by(year,sex) %>%
  count() %>%
  filter(n > 1)
```

```
## # A tibble: 0 x 3
## # Groups:   year, sex [0]
## # ... with 3 variables: year <int>, sex <chr>, n <int>
```

```
# babynames: the keys are year, sex, and name.
babynames %>%
  group_by(year, sex, name) %>%
  count() %>%
  filter(n > 1)
```

```
## # A tibble: 0 x 4
## # Groups:   year, sex, name [0]
## # ... with 4 variables: year <dbl>, sex <chr>, name <chr>, n <int>
```

```
# babynames: the keys are year, and births.
births %>%
  group_by(year, births) %>%
  count() %>%
  filter(n > 1)
```

```
## # A tibble: 0 x 3
## # Groups:   year, births [0]
## # ... with 3 variables: year <int>, births <int>, n <int>
```

```
# lifetables: the keys are x, sex, and year. (qx:ex are not)
lifetables %>%
  group_by(x,sex,year) %>%
  count() %>%
  filter(n > 1)
```

```
## Warning: Factor `sex` contains implicit NA, consider using
## `forcats::fct_explicit_na`
```

```
## # A tibble: 0 x 4
## # Groups:   x, sex, year [1]
## # ... with 4 variables: x <dbl>, sex <fct>, year <dbl>, n <int>
```

2. Read the description of the nasaweather dataset with (you might need to install nasaweather)

```
library(help = "nasaweather")
```

What are the data frames in this data set?

Ans: Atmospheric data, Country borders, Elevation, Glacier locations and Storm tracks data.

```
library(nasaweather)
```

```
##
## Attaching package: 'nasaweather'

## The following object is masked from 'package:dplyr':
##
##     storms
```

```
head(atmos) #Atmospheric_data.
```

```
## # A tibble: 6 x 11
##     lat  long  year month surftemp  temp pressure ozone cloudlow cloudmid
##   <dbl> <dbl> <int> <int>    <dbl> <dbl>    <dbl> <dbl>    <dbl>    <dbl>
## 1  36.2 -114.  1995     1     273.  272.      835   304      7.5     34.5
## 2  33.7 -114.  1995     1     280.  282.      940   304     11.5     32.5
## 3  31.2 -114.  1995     1     285.  285.      960   298     16.5     26
## 4  28.7 -114.  1995     1     289.  291.      990   276     20.5     14.5
## 5  26.2 -114.  1995     1     292.  293.     1000   274     26       10.5
## 6  23.7 -114.  1995     1     294.  294.     1000   264     30        9.5
## # ... with 1 more variable: cloudhigh <dbl>
```

```
head(borders) #Country_borders
```

```
## Warning: Detecting old grouped_df format, replacing `vars` attribute by `groups`

## # A tibble: 6 x 4
## # Groups:   group [1]
##   country  long   lat group
##   <chr>   <dbl> <dbl> <int>
## 1 AG      -61.7  17.0     1
## 2 AG      -61.7  17.0     1
## 3 AG      -61.9  17.0     1
## 4 AG      -61.9  17.1     1
## 5 AG      -61.9  17.1     1
## 6 AG      -61.8  17.2     1
```

```
head(elev) #Elevation.
```

```
## # A tibble: 6 x 3
##    long    lat  elev
##   <dbl>  <dbl> <dbl>
## 1 -114. -21.2      0
## 2 -114. -18.7      0
## 3 -114. -16.2      0
## 4 -114. -13.7      0
## 5 -114. -11.2      0
## 6 -114.  -8.72     0
```

```
head(glaciers) #Glacier_locations
```

```
## # A tibble: 6 x 6
##   id          name          lat  long area   country
##   <chr>       <chr>       <dbl> <dbl> <chr>  <chr>
## 1 C01A0101001 RAMIREZ E 4  10.8 -73.6 " NA"  CO
## 2 C01A0101002 RAMIREZ E 3  10.8 -73.6 " NA"  CO
## 3 C01A0101003 RAMIREZ E 2  10.8 -73.6 " NA"  CO
## 4 C01A0101004 RAMIREZ E 1  10.8 -73.6 "0.03" CO
## 5 C01A0101005 RAMIREZ 5 N  10.8 -73.6 "0.1"  CO
## 6 C01A0101007 RAMIREZ 3 N  10.8 -73.6 "0.03" CO
```

```
head(storms) #Storm_tracks_data
```

```
## # A tibble: 6 x 11
##   name    year month   day  hour   lat  long pressure  wind type          seasday
##   <chr>  <int> <int> <int> <int> <dbl> <dbl>    <int> <int> <chr>           <int>
## 1 Allis...  1995     6     3     0  17.4 -84.3     1005    30 Tropical De...       3
## 2 Allis...  1995     6     3     6  18.3 -84.9     1004    30 Tropical De...       3
## 3 Allis...  1995     6     3    12  19.3 -85.7     1003    35 Tropical St...       3
## 4 Allis...  1995     6     3    18  20.6 -85.8     1001    40 Tropical St...       3
## 5 Allis...  1995     6     4     0  22   -86        997    50 Tropical St...       4
## 6 Allis...  1995     6     4     6  23.3 -86.3      995    60 Tropical St...       4
```

```
# Test note chunk
if (FALSE){
  Atmospheric_data.
  Country_borders
  Elevation.
  Glacier_locations
  Storm_tracks_data

}
```

What are the keys in each data frame?

```
# atmos:the keys are lat, long, year, month
nasaweather::atmos %>%
  count(lat, long, year, month) %>%
  filter(n > 1) %>%
  nrow()
```

```
## [1] 0
```

```
# borders:No key in this data frame
nasaweather::borders %>%
  count(country, long, lat, group) %>%
  filter(n > 1) %>%
  nrow()
```

```
## [1] 154
```

```r
# elev:the keys are long, lat, elev
nasaweather::elev %>%
  count(long, lat, elev) %>%
  filter(n > 1) %>%
  nrow()
```

```
## [1] 0
```

```r
# glaciers:the key is id
nasaweather::glaciers %>%
  count(id) %>%
  filter(n > 1) %>%
  nrow()
```

```
## [1] 0
```

```r
# storms:the key is name,year, month, day, hour, lat, long
nasaweather::storms %>%
  count(name,year, month, day, hour, lat, long) %>%
  filter(n > 1) %>%
  nrow()
```

```
## [1] 0
```

Exercise 3: Lahman's Baseball Dataset

This exercise concerns the Lahman dataset. You can read about it with:

```r
library(tidyverse)
library(Lahman)
```

```
## Warning: package 'Lahman' was built under R version 3.6.3
```

```r
help("Lahman-package")
```

```
## starting httpd help server ... done
```

For this exercise, we'll use the Master, Batting, Pitching, Fielding, Teams, and Salaries data frames.

1. Load these data frames into R and read about them.

```r
head(Master)
```

```
##    playerID birthYear birthMonth birthDay birthCountry birthState  birthCity
## 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
##   deathYear deathMonth deathDay deathCountry deathState deathCity nameFirst
## 1        NA         NA       NA         <NA>       <NA>      <NA>     David
## 2        NA         NA       NA         <NA>       <NA>      <NA>      Hank
## 3      1984          8       16          USA         GA   Atlanta    Tommie
## 4        NA         NA       NA         <NA>       <NA>      <NA>       Don
## 5        NA         NA       NA         <NA>       <NA>      <NA>      Andy
## 6        NA         NA       NA         <NA>       <NA>      <NA>  Fernando
##   nameLast        nameGiven weight height bats throws      debut  finalGame
## 1  Aardsma      David Allan    215     75    R      R 2004-04-06 2015-08-23
## 2    Aaron      Henry Louis    180     72    R      R 1954-04-13 1976-10-03
## 3    Aaron       Tommie Lee    190     75    R      R 1962-04-10 1971-09-26
## 4     Aase   Donald William    190     75    R      R 1977-07-26 1990-10-03
## 5     Abad    Fausto Andres    184     73    L      L 2001-09-10 2006-04-13
## 6     Abad Fernando Antonio    220     73    L      L 2010-07-28 2017-10-01
##    retroID   bbrefID  deathDate  birthDate
## 1 aardd001 aardsda01       <NA> 1981-12-27
## 2 aaroh101 aaronha01       <NA> 1934-02-05
## 3 aarot101 aaronto01 1984-08-16 1939-08-05
## 4 aased001  aasedo01       <NA> 1954-09-08
## 5 abada001  abadan01       <NA> 1972-08-25
## 6 abadf001  abadfe01       <NA> 1985-12-17
```

```r
head(Batting)
```

```
##    playerID yearID stint teamID lgID  G  AB  R  H X2B X3B HR RBI SB CS BB SO
## 1 abercda01   1871     1    TRO   NA  1   4  0  0   0   0  0   0  0  0  0  0
## 2  addybo01   1871     1    RC1   NA 25 118 30 32   6   0  0  13  8  1  4  0
## 3 allisar01   1871     1    CL1   NA 29 137 28 40   4   5  0  19  3  1  2  5
## 4 allisdo01   1871     1    WS3   NA 27 133 28 44  10   2  2  27  1  1  0  2
## 5 ansonca01   1871     1    RC1   NA 25 120 29 39  11   3  0  16  6  2  2  1
## 6 armstbo01   1871     1    FW1   NA 12  49  9 11   2   1  0   5  0  1  0  1
##   IBB HBP SH SF GIDP
## 1  NA  NA NA NA    0
## 2  NA  NA NA NA    0
## 3  NA  NA NA NA    1
## 4  NA  NA NA NA    0
## 5  NA  NA NA NA    0
## 6  NA  NA NA NA    0
```

```r
head(Pitching)
```

```
##    playerID yearID stint teamID lgID  W  L  G GS CG SHO SV IPouts   H  ER HR BB
## 1 bechtge01   1871     1    PH1   NA  1  2  3  3  2   0  0     78  43  23  0 11
## 2 brainas01   1871     1    WS3   NA 12 15 30 30 30   0  0    792 361 132  4 37
## 3 fergubo01   1871     1    NY2   NA  0  0  1  0  0   0  0      3   8   3  0  0
## 4 fishech01   1871     1    RC1   NA  4 16 24 24 22   1  0    639 295 103  3 31
## 5 fleetfr01   1871     1    NY2   NA  0  1  1  1  1   0  0     27  20  10  0  3
## 6 flowedi01   1871     1    TRO   NA  0  0  1  0  0   0  0      3   1   0  0  0
##   SO BAOpp   ERA IBB WP HBP BK  BFP GF   R SH SF GIDP
## 1  1    NA  7.96  NA  7  NA  0  146  0  42 NA NA   NA
## 2 13    NA  4.50  NA  7  NA  0 1291  0 292 NA NA   NA
## 3  0    NA 27.00  NA  2  NA  0   14  0   9 NA NA   NA
## 4 15    NA  4.35  NA 20  NA  0 1080  1 257 NA NA   NA
## 5  0    NA 10.00  NA  0  NA  0   57  0  21 NA NA   NA
## 6  0    NA  0.00  NA  0  NA  0    3  1   0 NA NA   NA
```

```
head(Fielding)
```

```
##    playerID yearID stint teamID lgID POS  G GS InnOuts PO  A  E DP PB WP SB CS
## 1 abercda01   1871     1    TRO   NA  SS  1  1      24  1  3  2  0 NA NA NA NA
## 2  addybo01   1871     1    RC1   NA  2B 22 22     606 67 72 42  5 NA NA NA NA
## 3  addybo01   1871     1    RC1   NA  SS  3  3      96  8 14  7  0 NA NA NA NA
## 4 allisar01   1871     1    CL1   NA  2B  2  0      18  1  4  0  0 NA NA NA NA
## 5 allisar01   1871     1    CL1   NA  OF 29 29     729 51  3  7  1 NA NA NA NA
## 6 allisdo01   1871     1    WS3   NA   C 27 27     681 68 15 20  4 18 NA  0  0
##   ZR
## 1 NA
## 2 NA
## 3 NA
## 4 NA
## 5 NA
## 6 NA
```

```
head(Teams)
```

```
##   yearID lgID teamID franchID divID Rank  G Ghome  W  L DivWin WCWin LgWin
## 1   1871   NA    BS1      BNA  <NA>    3 31    NA 20 10   <NA>  <NA>     N
## 2   1871   NA    CH1      CNA  <NA>    2 28    NA 19  9   <NA>  <NA>     N
## 3   1871   NA    CL1      CFC  <NA>    8 29    NA 10 19   <NA>  <NA>     N
## 4   1871   NA    FW1      KEK  <NA>    7 19    NA  7 12   <NA>  <NA>     N
## 5   1871   NA    NY2      NNA  <NA>    5 33    NA 16 17   <NA>  <NA>     N
## 6   1871   NA    PH1      PNA  <NA>    1 28    NA 21  7   <NA>  <NA>     Y
##   WSWin   R   AB   H X2B X3B HR BB SO SB CS HBP SF  RA  ER  ERA CG SHO SV
## 1  <NA> 401 1372 426  70  37  3 60 19 73 16  NA NA 303 109 3.55 22   1  3
## 2  <NA> 302 1196 323  52  21 10 60 22 69 21  NA NA 241  77 2.76 25   0  1
## 3  <NA> 249 1186 328  35  40  7 26 25 18  8  NA NA 341 116 4.11 23   0  0
## 4  <NA> 137  746 178  19   8  2 33  9 16  4  NA NA 243  97 5.17 19   1  0
## 5  <NA> 302 1404 403  43  21  1 33 15 46 15  NA NA 313 121 3.72 32   1  0
## 6  <NA> 376 1281 410  66  27  9 46 23 56 12  NA NA 266 137 4.95 27   0  0
##   IPouts  HA HRA BBA SOA   E DP    FP                    name
## 1    828 367   2  42  23 243 24 0.834    Boston Red Stockings
## 2    753 308   6  28  22 229 16 0.829 Chicago White Stockings
## 3    762 346  13  53  34 234 15 0.818  Cleveland Forest Citys
## 4    507 261   5  21  17 163  8 0.803    Fort Wayne Kekiongas
## 5    879 373   7  42  22 235 14 0.840        New York Mutuals
## 6    747 329   3  53   16 194 13 0.845  Philadelphia Athletics
##                            park attendance BPF PPF teamIDBR teamIDlahman45
## 1          South End Grounds I         NA 103  98      BOS            BS1
## 2      Union Base-Ball Grounds         NA 104 102      CHI            CH1
## 3 National Association Grounds         NA  96 100      CLE            CL1
## 4               Hamilton Field         NA 101 107      KEK            FW1
## 5     Union Grounds (Brooklyn)         NA  90  88      NYU            NY2
## 6     Jefferson Street Grounds         NA 102  98      ATH            PH1
##   teamIDretro
## 1         BS1
## 2         CH1
## 3         CL1
## 4         FW1
## 5         NY2
## 6         PH1
```

```
head(Salaries)
```

```
##   yearID teamID lgID  playerID salary
## 1   1985    ATL   NL barkele01 870000
## 2   1985    ATL   NL bedrost01 550000
## 3   1985    ATL   NL benedbr01 545000
## 4   1985    ATL   NL  campri01 633333
## 5   1985    ATL   NL ceronri01 625000
## 6   1985    ATL   NL chambch01 800000
```

2. Find all the names of the players who have ever had a stint (from the Fielding data frame) in the Red Sox (or the Boston Americans) in years where they made it to the World Series (so they won their leagues).

- Show the first ten names (arranged in alphabetical order of last name).
- Note the World Series was not played each year and began in 1903.

```
Teams %>%
  filter(yearID >= 1903 ) %>%
  filter(LgWin == "Y" & teamID == "BOS") %>%
  filter(!is.na(WSWin)) %>% # WSWin should be Y or No
  select(yearID,lgID,teamID,LgWin,WSWin) -> Boston_League

left_join( Boston_League,Fielding,by = c("yearID", "lgID", "teamID")) %>%
  left_join(Master, by = "playerID" ) %>%
  filter(stint > 0) %>%
  select(nameFirst, nameLast, yearID) %>%
  arrange(nameLast) %>%
  distinct() %>% #unique() also
  head(10)
```

```
##    nameFirst nameLast yearID
## 1    Alfredo   Aceves   2013
## 2      Jerry    Adair   1967
## 3      Terry    Adams   2004
## 4        Sam    Agnew   1916
## 5        Sam    Agnew   1918
## 6       Nick  Altrock   1903
## 7        Abe  Alvarez   2004
## 8      Jimmy Anderson   2004
## 9      Ernie   Andres   1946
## 10       Kim   Andrew   1975
```

3. Some players play on multiple teams each year.

- Construct a data frame containing the total salary for each player for each year.
- Construct a second data frame containing the total number of at bats and hits for each player in a year.

```
Lahman::Salaries %>%
  group_by(yearID,playerID) %>%
  summarise(salary_total = sum(salary, na.rm = TRUE)) -> salary_total

Lahman::Batting %>%
  group_by(yearID,playerID) %>%
  summarise(total_bats = sum(AB, na.rm = TRUE),
            total_hits = sum(H, na.rm = TRUE)) -> total_bats_and_hits

full_join(salary_total, total_bats_and_hits, by = c("yearID", "playerID")) ->
multiple_teams_each_year

multiple_teams_each_year
```

```
## # A tibble: 98,815 x 5
## # Groups:   yearID [148]
##    yearID playerID  salary_total total_bats total_hits
##     <int> <chr>            <int>      <int>      <int>
##  1   1985 ackerji01       170000          0          0
##  2   1985 agostju01       147500          0          0
##  3   1985 aguaylu01       237000        165         46
##  4   1985 alexado01       875000          0          0
##  5   1985 allenne01       750000          2          0
##  6   1985 almonbi01       255000        244         66
##  7   1985 anderal02        62500         NA         NA
##  8   1985 anderla02       250500          4          0
##  9   1985 andujjo01      1030000         94         10
## 10   1985 armasto01       915000        385        102
## # ... with 98,805 more rows
```

4. The batting average of a player is the number of Hits divided by the number of at bats. A larger value is good.

- Using the data frames you created in part 3, create a new data frame with batting average and salary information for only players in the years after 1985 (when salary information started being collected) who had a minimum of 400 at bats.

```
after_1985 <- multiple_teams_each_year %>%
  mutate(batting_average = (total_hits / total_bats)) %>%
  filter(yearID > 1985 & total_bats >= 400) %>% #including 1985?
  select(yearID, playerID, batting_average, salary_total)

after_1985
```

```
## # A tibble: 5,681 x 4
## # Groups:   yearID [33]
##    yearID playerID  batting_average salary_total
##     <int> <chr>               <dbl>        <int>
##  1   1986 armasto01           0.264      1000000
##  2   1986 baineha01           0.296       775000
##  3   1986 balbost01           0.229       525000
##  4   1986 barfije01           0.289       725000
##  5   1986 barrema02           0.286       435000
##  6   1986 basske01            0.311       310000
##  7   1986 baylodo01           0.238       660696
##  8   1986 bellbu01            0.278       870000
##  9   1986 bellge02            0.309       725000
## 10   1986 bernato01           0.301       580000
## # ... with 5,671 more rows
```

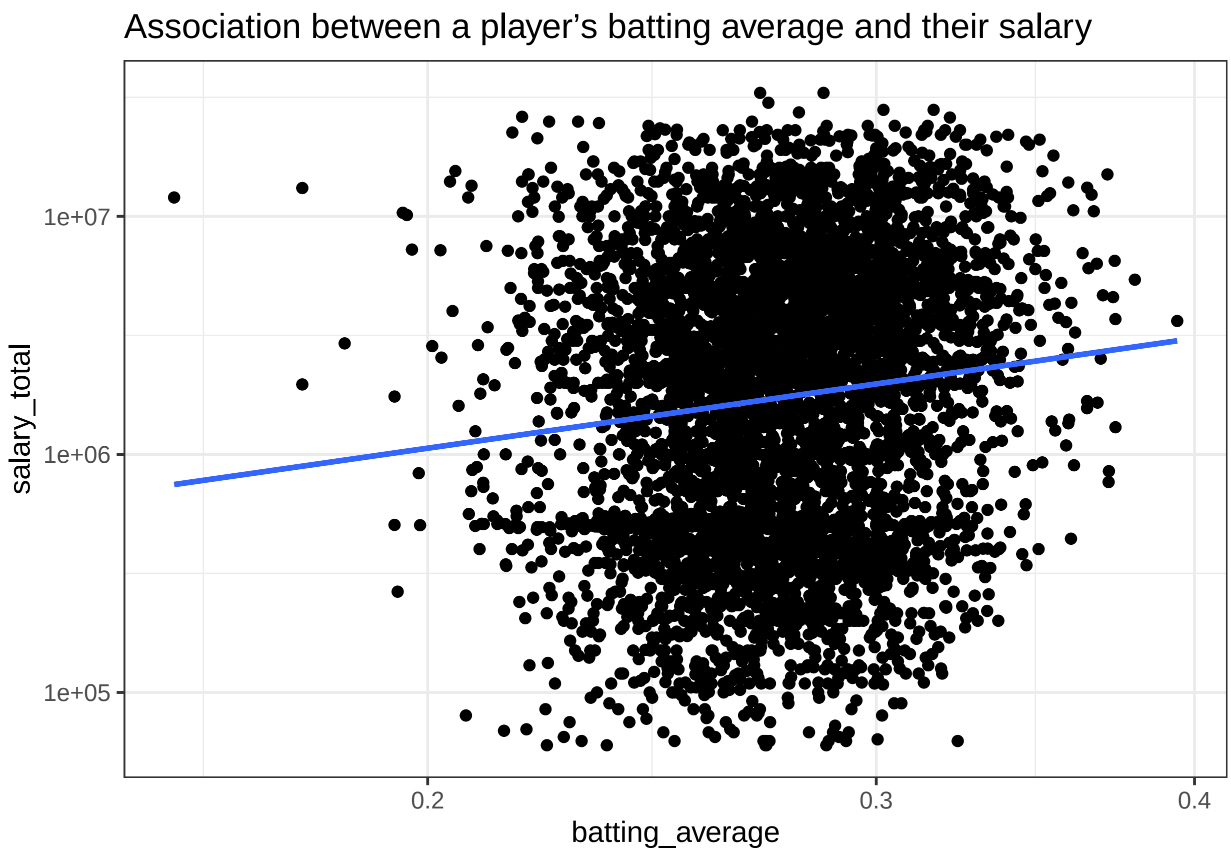
- Explore the marginal association between a player's batting average and their salary.

```
after_1985 %>%
  ggplot(aes(x = batting_average, y = salary_total))+
  geom_point()+
  scale_x_log10()+
  scale_y_log10()+
  ggtitle("Association between a player's batting average and their salary") +
  geom_smooth(method = "lm", se = FALSE )+
  theme_bw()
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 492 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 492 rows containing missing values (geom_point).
```



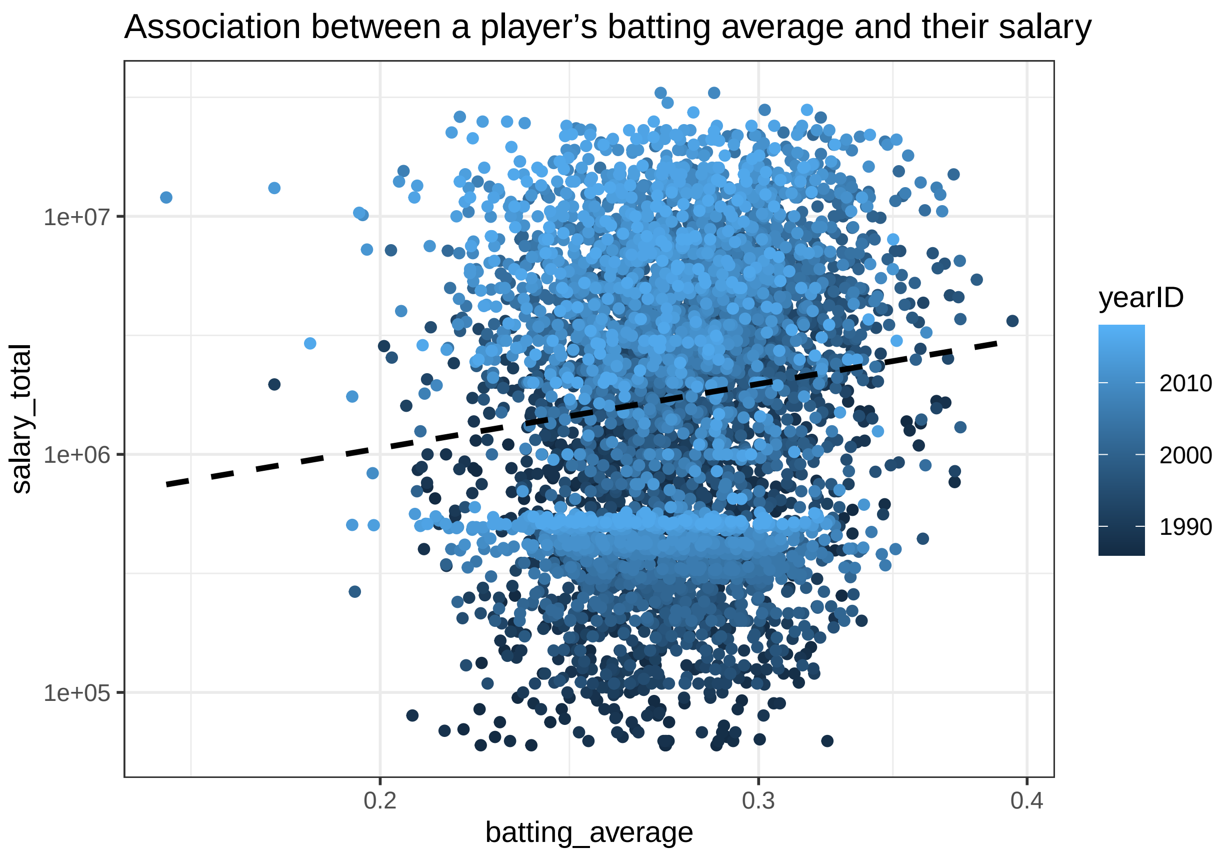
- Explore if this association has changed over time (for example, because sports teams are getting more stats-savvy). Hint: figure out how to set the color based on year.

```
after_1985 %>%
  ggplot(aes(x = batting_average, y = salary_total, color = yearID ))+
  geom_point()+
  scale_x_log10()+
  scale_y_log10()+
  ggtitle("Association between a player's batting average and their salary") +
  geom_smooth(method = "lm", se = FALSE, color = "black",linetype = "dashed")+
  theme_bw()
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 492 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 492 rows containing missing values (geom_point).
```



Conclusion: The relationship between the batting average and the total salary is positive. On the other hand, when the batting average is increasing, the total salary is increasing, too. (the year of legend is not affected the batting average and total of salary.) Therefore, what happens to totally salary, while batting average is increasing over time.

5. Find the salary of all players named "John" in even numbered years after 1985. Print the first ten values arranged in descending order of salary.

```
Lahman::Master %>%
filter(nameFirst == "John") -> John_bio


left_join(John_bio, Salaries, by = "playerID") %>%
  select(yearID,nameFirst,nameLast,salary) %>%
  arrange(desc(salary)) %>%
  filter(yearID %%2 == 0) %>%
  head(10)
```

```
##    yearID nameFirst nameLast   salary
## 1    2010      John   Lackey 18700000
## 2    2016      John   Lackey 16000000
## 3    2012      John   Lackey 15950000
## 4    2016      John    Danks 15750000
## 5    2014      John   Lackey 15250000
## 6    2014      John    Danks 14250000
## 7    2008      John   Smoltz 14000000
## 8    2004      John   Smoltz 11666667
## 9    2006      John   Smoltz 11000000
## 10   2000      John   Smoltz  8500000
```