

MA388 Sabermetrics: Lesson 3

Introduction to R - Part II

```
library(Lahman)
library(tidyverse)
library(knitr)
```

Review

Key Concepts

- five tidyverse verbs
- relational databases

Review Questions

- Using the **Teams** data frame, calculate average wins (W) per season for each MLB team from 2000-present. Order your results from the best teams to the worst.

| teamID | W | G | Wpct |
|--------|------|------|-------|
| NYA | 2286 | 3946 | 0.579 |
| LAN | 2239 | 3948 | 0.567 |
| SLN | 2179 | 3945 | 0.552 |
| ATL | 2152 | 3945 | 0.546 |
| BOS | 2145 | 3947 | 0.543 |
| ANA | 425 | 810 | 0.525 |
| SFN | 2052 | 3946 | 0.520 |
| CLE | 2046 | 3945 | 0.519 |
| OAK | 2030 | 3946 | 0.514 |
| HOU | 2029 | 3947 | 0.514 |
| LAA | 1609 | 3138 | 0.513 |
| PHI | 2019 | 3947 | 0.512 |

| teamID | W | G | Wpct |
|--------|------|------|-------|
| MIN | 1984 | 3949 | 0.502 |
| TOR | 1977 | 3947 | 0.501 |
| NYN | 1976 | 3946 | 0.501 |
| SEA | 1975 | 3948 | 0.500 |
| CHN | 1969 | 3947 | 0.499 |
| TBA | 1959 | 3946 | 0.496 |
| MIL | 1960 | 3949 | 0.496 |
| FLO | 963 | 1942 | 0.496 |
| TEX | 1937 | 3949 | 0.491 |
| ARI | 1922 | 3948 | 0.487 |
| CHA | 1911 | 3948 | 0.484 |
| WAS | 1510 | 3136 | 0.482 |
| SDN | 1888 | 3949 | 0.478 |
| CIN | 1861 | 3950 | 0.471 |
| DET | 1841 | 3943 | 0.467 |
| COL | 1809 | 3949 | 0.458 |
| BAL | 1805 | 3948 | 0.457 |
| MON | 368 | 810 | 0.454 |
| PIT | 1767 | 3944 | 0.448 |
| KCA | 1737 | 3948 | 0.440 |
| MIA | 868 | 2002 | 0.434 |

- Using the `Teams` data frame, calculate how many World Series titles (WSWin) each team had during this period and add this information to your table in the previous question.

Table 2: Win Percentage and World Series Titles by Team (2000-present).

| teamID | W | G | Wpct | WSWins |
|--------|------|------|-------|--------|
| NYA | 2286 | 3946 | 0.579 | 2 |
| LAN | 2239 | 3948 | 0.567 | 2 |
| SLN | 2179 | 3945 | 0.552 | 2 |
| ATL | 2152 | 3945 | 0.546 | 1 |
| BOS | 2145 | 3947 | 0.543 | 4 |
| ANA | 425 | 810 | 0.525 | 1 |
| SFN | 2052 | 3946 | 0.520 | 3 |
| CLE | 2046 | 3945 | 0.519 | 0 |
| OAK | 2030 | 3946 | 0.514 | 0 |
| HOU | 2029 | 3947 | 0.514 | 2 |
| LAA | 1609 | 3138 | 0.513 | 0 |
| PHI | 2019 | 3947 | 0.512 | 1 |
| MIN | 1984 | 3949 | 0.502 | 0 |

| teamID | W | G | Wpct | WSWins |
|--------|------|------|-------|--------|
| TOR | 1977 | 3947 | 0.501 | 0 |
| NYN | 1976 | 3946 | 0.501 | 0 |
| SEA | 1975 | 3948 | 0.500 | 0 |
| CHN | 1969 | 3947 | 0.499 | 1 |
| TBA | 1959 | 3946 | 0.496 | 0 |
| MIL | 1960 | 3949 | 0.496 | 0 |
| FLO | 963 | 1942 | 0.496 | 1 |
| TEX | 1937 | 3949 | 0.491 | 1 |
| ARI | 1922 | 3948 | 0.487 | 1 |
| CHA | 1911 | 3948 | 0.484 | 1 |
| WAS | 1510 | 3136 | 0.482 | 1 |
| SDN | 1888 | 3949 | 0.478 | 0 |
| CIN | 1861 | 3950 | 0.471 | 0 |
| DET | 1841 | 3943 | 0.467 | 0 |
| COL | 1809 | 3949 | 0.458 | 0 |
| BAL | 1805 | 3948 | 0.457 | 0 |
| MON | 368 | 810 | 0.454 | 0 |
| PIT | 1767 | 3944 | 0.448 | 0 |
| KCA | 1737 | 3948 | 0.440 | 1 |
| MIA | 868 | 2002 | 0.434 | 0 |

Introduction to R (Day 2)

Split, Apply, and Combine Data

It's important in data science to be able to write procedures and apply them over a data set. The Marchi text refers to this as “splitting, applying, and combining data.” In other words, we split a data frame into pieces, apply a procedure or function to each piece, and then combine the results into a new data frame.

Ways to “split, apply, and combine”...

- `group_by()` and `summarize()` functions
- `split()` and `map_df()` functions

In general, these coding tasks involve four steps:

1. Write a function to perform the task on one split of the data.
2. Pull the individual values of the variable you are splitting on.
3. Split the data on a variable.
4. Apply the function to each split of the data.

Example

Find the team with the most wins for each season and determine whether the team won the World Series.

Four steps:

1. Write a function that takes a data frame and returns the team with the most wins and whether they won the World Series.
2. Pull the `yearID`s you will split on.
3. Split the data frame on `year`.
4. Apply the function to each split.

```
# Step 1 - Write a function.
mostWins <- function(data){
  data |>
    arrange(-W) |>
    select(teamID, W, WSWin) |>
    head(1)
}

# Step 2 - Pull the years you will split on.
year <- Teams |>
  filter(yearID >= 2000) |>
  group_by(yearID) |>
  group_keys() |>
  pull(yearID)

# Steps 3 and 4 - Split and apply.
winLeaders <- Teams |>
  filter(yearID >= 2000) |>
  split(year) |>
  map_df(mostWins, .id = "yearID")

winLeaders |>
  kable(caption = "Major League win leaders by season and whether they won the World Series.")
```

Table 3: Major League win leaders by season and whether they won the World Series.

| yearID | teamID | W | WSWin |
|--------|--------|----|-------|
| 2000 | SFN | 97 | N |

| yearID | teamID | W | WSWin |
|--------|--------|-----|-------|
| 2001 | SEA | 116 | N |
| 2002 | NYA | 103 | N |
| 2003 | ATL | 101 | N |
| 2004 | SLN | 105 | N |
| 2005 | SLN | 100 | N |
| 2006 | NYN | 97 | N |
| 2007 | BOS | 96 | Y |
| 2008 | LAA | 100 | N |
| 2009 | NYA | 103 | Y |
| 2010 | PHI | 97 | N |
| 2011 | PHI | 102 | N |
| 2012 | WAS | 98 | N |
| 2013 | BOS | 97 | Y |
| 2014 | LAA | 98 | N |
| 2015 | SLN | 100 | N |
| 2016 | CHN | 103 | Y |
| 2017 | LAN | 104 | N |
| 2018 | BOS | 108 | Y |
| 2019 | HOU | 107 | N |
| 2020 | LAN | 43 | Y |
| 2021 | SFN | 107 | N |
| 2022 | LAN | 111 | N |
| 2023 | ATL | 104 | N |
| 2024 | LAN | 98 | Y |

Note: Functions are not executed until they are called.

Your Turn

Find the team with the most home runs for each season from 2000-present. Include the team's full name in your table instead of just the teamID.

Table 4: Major League win leaders by season and whether they won the World Series.

| yearID | name | HR |
|--------|-------------------|-----|
| 2000 | Houston Astros | 249 |
| 2001 | Texas Rangers | 246 |
| 2002 | Texas Rangers | 230 |
| 2003 | Texas Rangers | 239 |
| 2004 | Chicago White Sox | 242 |
| 2005 | Texas Rangers | 260 |

| yearID | name | HR |
|--------|---------------------|-----|
| 2006 | Chicago White Sox | 236 |
| 2007 | Milwaukee Brewers | 231 |
| 2008 | Chicago White Sox | 235 |
| 2009 | New York Yankees | 244 |
| 2010 | Toronto Blue Jays | 257 |
| 2011 | New York Yankees | 222 |
| 2012 | New York Yankees | 245 |
| 2013 | Baltimore Orioles | 212 |
| 2014 | Baltimore Orioles | 211 |
| 2015 | Toronto Blue Jays | 232 |
| 2016 | Baltimore Orioles | 253 |
| 2017 | New York Yankees | 241 |
| 2018 | New York Yankees | 267 |
| 2019 | Minnesota Twins | 307 |
| 2020 | Los Angeles Dodgers | 118 |
| 2021 | Toronto Blue Jays | 262 |
| 2022 | New York Yankees | 254 |
| 2023 | Atlanta Braves | 307 |
| 2024 | New York Yankees | 237 |

What's the difference between `select()` and `pull()`? One key difference is they return different data types.

```
HRLeaders |>
  select(yearID)
```

```
   yearID
1    2000
2    2001
3    2002
4    2003
5    2004
6    2005
7    2006
8    2007
9    2008
10   2009
11   2010
12   2011
13   2012
14   2013
15   2014
16   2015
```

```
17 2016
18 2017
19 2018
20 2019
21 2020
22 2021
23 2022
24 2023
25 2024
```

```
HRLeaders |>
  pull(yearID)
```

```
[1] "2000" "2001" "2002" "2003" "2004" "2005" "2006" "2007" "2008" "2009"
[11] "2010" "2011" "2012" "2013" "2014" "2015" "2016" "2017" "2018" "2019"
[21] "2020" "2021" "2022" "2023" "2024"
```