**Learning Goals**

Upon completion of this module students should improve upon

* being able to join two data tables
* being able to anticipate (by sketching) the resulting data table prior to running the code to do so
* identify situations where more than one key is needed

**Introduction**

Data for a particular sport is often stored across numerous locations. For example, in NCAA Division I Softball, batting statistics for each season are typically stored in separate tables. (See for example the statistics hosted by <https://d1softball.com/>)

Suppose we are interested in tracking the statistics of players across multiple seasons. A common way to prep the data to do this is to use join statements to merge each season’s data into one table with one row per player (and columns associated with their different statistics for each season).

This module looks at some simple batting stats over two seasons through the use of joining functions for a small subset of NCAA Division 1 Softball players’ statistics for the 2021 and 2022 seasons. (This is only a small window of a much bigger dataset).

**Data**

The data displayed below represent non-random samples taken the full data. **R** is the number of runs scored by the player, **H** is the number of hits. These subsets, as well as the full data (batting2021.csv and batting2022.csv) are available for download on the GitHub repo associated with this module. The full data also includes additional batting statistics.

**batting2021\_subset**

|  |  |  |  |
| --- | --- | --- | --- |
| **Player** | **Team** | **R** | **H** |
| Aaliyah Swan | Cal State Northridge | 8 | 20 |
| Abbey Latham | Ole Miss | 25 | 53 |
| Bella Rocco | Boise State | 16 | 42 |
| Carson Fischer | Northern Colorado | 11 | 15 |
| Drew Dudley | Austin Peay | 10 | 29 |
| Emily Gant | Boston University | 31 | 36 |

**batting2022\_subset**

|  |  |  |  |
| --- | --- | --- | --- |
| **Player** | **Team** | **R** | **H** |
| Aaliyah Swan | Cal State Northridge | 16 | 23 |
| Abbey Latham | Ole Miss | 36 | 45 |
| Bella Rocco | Boise State | 10 | 29 |
| Emily Gant | Boston University | 40 | 55 |
| Lexi Osowski | Austin Peay | 42 | 64 |
| Mikayla Allee | Ole Miss | 36 | 28 |

**Methods**

For R users, recall that six commonly used joining functions available through the dplyr package are, left, right, full, inner, semi, and anti. Additionally, other useful dplyr verbs for combining multiple tables of data include intersect, union, and setdiff. (These are called set operators.)

Further details regarding joins can be found at <https://dplyr.tidyverse.org/articles/two-table.html>.

**Exercises**

The first five exercises use the subset data previously displayed.

1. When using a join function on the datasets above, which variable would you want to use as your key and why?

*The variable you would want to use is Player and not team because there are multiple players on Ole Miss and there are two players from Austin Peay, but they aren’t the same players.*

1. left\_join batting2022\_subset to batting2021\_subset.
   1. Are you keeping the player names from batting2022\_subset or batting2021\_subset?

*We keep the player names from batting2021 and fill in the rest of the non-matching keys with NA.*

* 1. Sketch the dataset. If you have access to the appropriate technology, conduct the left\_join and check the output with your sketch.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Player*** | ***Team*** | ***R.x*** | ***H.x*** | ***R.y*** | ***H.y*** |
| *Aaliyah Swan* | *Cal State Northridge* | *8* | *20* | *16* | *23* |
| *Abbey Latham* | *Ole Miss* | *25* | *53* | *36* | *45* |
| *Bella Rocco* | *Boise State* | *16* | *42* | *10* | *29* |
| *Carson Fischer* | *Northern Colorado* | *11* | *15* | *NA* | *NA* |
| *Drew Dudley* | *Austin Peay* | *10* | *29* | *NA* | *NA* |
| *Emily Gant* | *Boston University* | *31* | *36* | *40* | *55* |

1. Would the dataset created by a right join of batting2022\_subset to batting2021\_subset be identical to the

dataset created in question 2? Why or why not?

*It would not be identical because the left\_join keeps the “left” dataset which is batting2021\_subset. The code for the right\_join would look similar, but it keeps the “right” dataset players which would be from batting2022\_subset. More specifically, you would instead have the rows for the six players from 2022 with NA in the 2021 R and H columns for Lexi Osowski and Mikayla Allee.*

1. If we wanted to see if players' performances improved or not from the 2021 season to the 2022 season, which of the join functions would be the most practical? Explain.

*We would need to use a join that ensures that we have the batting statistics for players that are in both seasons are joined. While multiple joins could work, the ideal join would likely be an inner join.*

1. Looking at the new table created below, provide the command that would result in this table.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Player** | **Team** | **R.x** | **H.y** | **R.x** | **H.y** |
| Aaliyah Swan | Cal State Northridge | 8 | 20 | 16 | 23 |
| Abbey Latham | Ole Miss | 25 | 53 | 36 | 45 |
| Bella Rocco | Boise State | 16 | 42 | 10 | 29 |
| Emily Gant | Boston University | 31 | 36 | 40 | 55 |

*An inner\_join was used because its only players with matching keys and it’s the variables from both datasets.*

*More specifically, using dplyr syntax:*

*inner\_join(x = batting2021\_subset, y = batting2022\_subset, by = “Player”)*

The remaining exercises use the full datasets. If you have the appropriate technology, check your answer for each by running the appropriate join.

1. Although not seen in the subset of data we used in the first five exercises, when considering multiple seasons of data for all NCAA Division 1 softball players, there is a reasonable chance that two players from different teams will have the same name. When using a join function, what would you want to use as your key? For now, let’s assume that players from the same team won’t have the same name and we’ll not concern ourselves with transfer students.

*You would need to use two keys, the player name and team, to uniquely identify each row.*

1. Suppose you only used the player’s name as the key, explain what would happen to your dataset if you did a full join?

*You could end up mistakenly matching too many rows to each other. (e.g., Investigate Kaitlyn Tucker. If you only use player as you key, you will have 4 rows instead of just two.)*

1. What is the difference between a full join and inner join? Answer based on what type of players would be included in each dataset.

*A full join will create a dataset that has rows for every player that has played in either 2021 or 2022, while the inner join will keep only those players that play in both seasons.*

1. Describe the dataset you would create when using an anti­­ join of batting2022 to batting2021. Provide context as to what this represents for college softball. If you have the appropriate technology, check your answer by running each of these joins.

*This would represent the players that from 2022 that did not play in 2021 (keeping only the 2022 statistics). An example would be first year college students.*

1. How would that differ if you used an anti join of batting2021 to batting2022? Provide context as to what this represents for college softball. If you have the appropriate technology, check your answer by running each of these joins.

*This would represent the players that from 2021 that did not play in 2022 (keeping only the 2021 statistics). An example would be Seniors in 2021 that no longer had NCAA eligibility.*

1. Suppose you want to analyze trends batting averages across 10 seasons of data. You should assume that all 10 seasons have the same column names.
   1. Briefly explain why repeatedly using an inner join to merge all 10 seasons would not be a good approach.

*Your dataset would not have any players in it as inner join would only keep players that have played all ten seasons. (Which is impossible under NCAA rules.)*

* 1. Briefly explain why repeatedly using use a full join to merge all 10 seasons would not be an ideal approach.

*While a full join would collect rows for all the players, joining would result in a huge number of blank (NA) entries. This would also (most likely) require that you pivot the data to make it suitable for analysis.*

* 1. What would be a better solution?

*A set operator that stacks the data would be a better solution. Examples would be union or bind\_rows.*