**Assessment: Combining Tables**

**Question 1**

1/1 point (graded)

You have created data frames tab1 and tab2 of state population and election data, similar to our module videos:

> tab1

state population

Alabama 4779736

Alaska 710231

Arizona 6392017

Delaware 897934

District of Columbia 601723

> tab2

state electoral\_votes

Alabama 9

Alaska 3

Arizona 11

California 55

Colorado 9

Connecticut 7

> dim(tab1)

[1] 5 2

> dim(tab2)

[1] 6 2

What are the dimensions of the table dat, created by the following command?

dat <- left\_join(tab1, tab2, by = “state”)

3 rows by 3 columns

5 rows by 2 columns

5 rows by 3 columns

6 rows by 3 columns

correct

Answer

Correct:

When we use a left\_join command, all rows in the left-hand table (in this case, tab1) are retained in the final table, so we expect to have five rows. In addition, columns from both tables will be included in the final “dat” table so we expect to have three columns.

You have used 1 of 2 attempts Some problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

**Question 2**

1/1 point (graded)

We are still using the tab1 and tab2 tables shown in question 1. What join command would create a new table “dat” with three rows and two columns?

dat <- right\_join(tab1, tab2, by = “state”)

dat <- full\_join(tab1, tab2, by = “state”)

dat <- inner\_join(tab1, tab2, by = “state”)

dat <- semi\_join(tab1, tab2, by = “state”)

correct

Answer

Correct:

The semi\_join command takes tab1 and limits it to states that are also in tab2, without adding the additional columns in tab2. This gives us three rows (states in both tables) and two columns (state and population, the two columns in tab1).

You have used 1 of 2 attempts Some problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

**Question 3**

1/1 point (graded)

Which of the following are real differences between the join and bind functions?

Please select all correct answers.

Binding functions combine by position, while join functions match by variables.

Joining functions can join datasets of different dimensions, but the bind functions must match on the appropriate dimension (either same row or column numbers).

Bind functions can combine both vectors and dataframes, while join functions work for only for dataframes.

The join functions are a part of the dplyr package and have been optimized for speed, while the bind functions are inefficient base functions.

correct

You have used 1 of 2 attempts Some problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

**Question 4**

1/1 point (graded)

We have two simple tables, shown below, with columns x and y:

> df1

x y

a a

b a

> df2

x y

a a

a b

Which command would result in the following table?

> final

x y

b a

final <- union(df1, df2)

final <- setdiff(df1, df2)

final <- setdiff(df2, df1)

final <- intersect(df1, df2)

correct

Answer

Correct:

The setdiff command returns rows in df1 but not df2, which matches our table final.

You have used 1 of 2 attempts Some problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

**Introduction to Questions 5-7**

Install and load the **Lahman** library. This library contains a variety of datasets related to US professional baseball. We will use this library for the next few questions and will discuss it more extensively in the Regression course. For now, focus on wrangling the data rather than understanding the statistics.

The **Batting** data frame contains the offensive statistics for all baseball players over several seasons.  Filter this data frame to define **top** as the top 10 home run (**HR**) hitters in 2016:

library(Lahman)

top <- Batting %>%

filter(yearID == 2016) %>%

arrange(desc(HR)) %>% # arrange by descending HR count

slice(1:10) # take entries 1-10

top %>% as\_tibble()

Also Inspect the **Master** data frame, which has demographic information for all players:

Master %>% as\_tibble()

**Question 5**

1/1 point (graded)

Use the correct join or bind function to create a combined table of the names and statistics of the top 10 home run (HR) hitters for 2016. This table should have the player ID, first name, last name, and number of HR for the top 10 players. Name this data frame top\_names.

Identify the join or bind that fills the blank in this code to create the correct table:

top\_names <- top %>% \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ %>%

select(playerID, nameFirst, nameLast, HR)

Which bind or join function fills the blank to generate the correct table?

rbind(Master)

cbind(Master)

left\_join(Master)

right\_join(Master)

full\_join(Master)

anti\_join(Master)

correct

You have used 1 of 2 attempts Some problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

**Question 6**

1/1 point (graded)

Inspect the Salaries data frame. Filter this data frame to the 2016 salaries, then use the correct bind join function to add a salary column to the top\_names data frame from the previous question. Name the new data frame top\_salary. Use this code framework:

top\_salary <- Salaries %>% filter(yearID == 2016) %>%

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ %>%

select(nameFirst, nameLast, teamID, HR, salary)

Which bind or join function fills the blank to generate the correct table?

rbind(top\_names)

cbind(top\_names)

left\_join(top\_names)

right\_join(top\_names)

full\_join(top\_names)

anti\_join(top\_names)

correct

You have used 1 of 2 attempts Some problems have options such as save, reset, hints, or show answer. These options follow the Submit button.

**Question 7**

2/2 points (graded)

Inspect the AwardsPlayers table. Filter awards to include only the year 2016.

How many players from the top 10 home run hitters won at least one award in 2016?

Use a set operator.

correct

3 Loading

How many players won an award in 2016 but were not one of the top 10 home run hitters in 2016?

Use a set operator.

correct

44 Loading

You have used 4 of 10 attempts