Joining Data Tables

Luna L Sanchez Reyes

2023-03-14

Use the shortcut to add a code block `ctrl + option + i` on mac `ctrl + alt i` on Windows.

Load the three data sets that we are going to join, surveys.csv, species.csv, plots.csv:

```
surveys <- read.csv(file = "../data-raw/surveys.csv")
species <- read.csv(file = "../data-raw/species.csv")
plots <- read.csv(file = "../data-raw/plots.csv")
```

Why do we need to combine or join data tables

Homework: elaborate on this topic

How de we join data tables in R

There is a group of functions `_join()` that allow us to combine two data tables using values on a shared column.

There has to be a shared column; and we need three main arguments to run these functions, two data tables and one column name that has to be provided as a character value.

The different functions allow us to combine in different ways.

We can run `inner_join` in the classic way:

```
inner_join(surveys, species, by = "species_id")
```

We can also run it using pipes:

```
surveys %>%
  inner_join(species, by = "species_id") -> joined_table
```

How can we explore our combined/joined table?

We want to see the differences between the two input tables and the resulting table. To see the differences in columns, we can use `head()`:

```
head(species)
```

```
##   species_id           genus         species   taxa
## 1         AB       Amphispiza       bilineata   Bird
## 2         AH Ammospermophilus         harrisi Rodent
## 3         AS       Ammodramus      savannarum   Bird
## 4         BA          Baiomys         taylori Rodent
## 5         CB  Campylorhynchus brunneicapillus   Bird
## 6         CM      Calamospiza     melanocorys   Bird
```

```
head(surveys)
```

```
##   record_id month day year plot_id species_id sex hindfoot_length weight
## 1         1     7  16 1977       2         NL   M              32     NA
## 2         2     7  16 1977       3         NL   M              33     NA
## 3         3     7  16 1977       2         DM   F              37     NA
## 4         4     7  16 1977       7         DM   M              36     NA
## 5         5     7  16 1977       3         DM   M              35     NA
## 6         6     7  16 1977       1         PF   M              14     NA
```

```
head(joined_table)
```

```
##   record_id month day year plot_id species_id sex hindfoot_length weight
## 1         1     7  16 1977       2         NL   M              32     NA
## 2         2     7  16 1977       3         NL   M              33     NA
## 3         3     7  16 1977       2         DM   F              37     NA
## 4         4     7  16 1977       7         DM   M              36     NA
## 5         5     7  16 1977       3         DM   M              35     NA
## 6         6     7  16 1977       1         PF   M              14     NA
##         genus  species   taxa
## 1     Neotoma albigula Rodent
## 2     Neotoma albigula Rodent
## 3   Dipodomys merriami Rodent
## 4   Dipodomys merriami Rodent
## 5   Dipodomys merriami Rodent
## 6 Perognathus   flavus Rodent
```

To explore the differences in numbers of rows, we can use the `str()` function:

```
str(species)
```

```
## 'data.frame':    54 obs. of  4 variables:
##  $ species_id: chr  "AB" "AH" "AS" "BA" ...
##  $ genus     : chr  "Amphispiza" "Ammospermophilus" "Ammodramus" "Baiomys" ...
##  $ species   : chr  "bilineata" "harrisi" "savannarum" "taylori" ...
##  $ taxa      : chr  "Bird" "Rodent" "Bird" "Rodent" ...
```

```
str(surveys)
```

```
## 'data.frame':    35549 obs. of  9 variables:
##  $ record_id      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ month          : int  7 7 7 7 7 7 7 7 7 7 ...
##  $ day            : int  16 16 16 16 16 16 16 16 16 16 ...
##  $ year           : int  1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 ...
##  $ plot_id        : int  2 3 2 7 3 1 2 1 1 6 ...
##  $ species_id     : chr  "NL" "NL" "DM" "DM" ...
##  $ sex            : chr  "M" "M" "F" "M" ...
##  $ hindfoot_length: int  32 33 37 36 35 14 NA 37 34 20 ...
##  $ weight         : int  NA NA NA NA NA NA NA NA NA NA ...
```

```
str(joined_table)
```

```
## 'data.frame':    34786 obs. of  12 variables:
##  $ record_id      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ month          : int  7 7 7 7 7 7 7 7 7 7 ...
##  $ day            : int  16 16 16 16 16 16 16 16 16 16 ...
##  $ year           : int  1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 ...
```

```
##  $ plot_id        : int  2 3 2 7 3 1 2 1 1 6 ...
##  $ species_id     : chr  "NL" "NL" "DM" "DM" ...
##  $ sex            : chr  "M" "M" "F" "M" ...
##  $ hindfoot_length: int  32 33 37 36 35 14 NA 37 34 20 ...
##  $ weight         : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ genus          : chr  "Neotoma" "Neotoma" "Dipodomys" "Dipodomys" ...
##  $ species        : chr  "albigula" "albigula" "merriami" "merriami" ...
##  $ taxa           : chr  "Rodent" "Rodent" "Rodent" "Rodent" ...
```

What happened with the number of rows in joined_table vs surveys?

It dropped the rows that did not have matching values of the species_id column

Exercise 1

Use `inner_join()` and `filter()` to get a data frame with the information from the `surveys` and `plots` tables where the "plot_type" is "Control".

```
surveys %>%
  inner_join(plots, by = "plot_type")
```

```
## Error in `inner_join()`:
## ! Join columns in `x` must be present in the data.
## x Problem with `plot_type`.
```

This returns an error because we tried to join by a column that is not shared by both data tables.

unique(plots$plot_type)

```
surveys %>%
  inner_join(plots, by = "plot_id") %>%
  filter(plot_type == "Control") %>%
  head()
```

```
##   record_id month day year plot_id species_id sex hindfoot_length weight
## 1         1     7  16 1977       2         NL   M              32     NA
## 2         3     7  16 1977       2         DM   F              37     NA
## 3         7     7  16 1977       2         PE   F              NA     NA
## 4        14     7  16 1977       8         DM                  NA     NA
## 5        16     7  16 1977       4         DM   F              36     NA
## 6        18     7  16 1977       2         PP   M              22     NA
##   plot_type
## 1   Control
## 2   Control
## 3   Control
## 4   Control
## 5   Control
## 6   Control
```

Automate joining tables and other things with `intersect()`

Which species_id values are shared between the two data tables

```
intersect(surveys$species_id, species$species_id)
```

```
##  [1] "NL" "DM" "PF" "PE" "DS" "PP" "SH" "OT" "DO" "OX" "SS" "OL" "RM" "SA" "PM"
## [16] "AH" "DX" "AB" "CB" "CM" "CQ" "RF" "PC" "PG" "PH" "PU" "CV" "UR" "UP" "ZL"
## [31] "UL" "CS" "SC" "BA" "SF" "RO" "AS" "SO" "PI" "ST" "CU" "SU" "RX" "PB" "PL"
## [46] "PX" "CT" "US"
```

To find shared columns we use the `colnames()` function:

```
colnames(surveys)
```

```
## [1] "record_id"       "month"           "day"             "year"
## [5] "plot_id"         "species_id"      "sex"             "hindfoot_length"
## [9] "weight"
```

```
colnames(species)
```

```
## [1] "species_id" "genus"      "species"    "taxa"
```

```
intersect(colnames(surveys), colnames(species))
```

```
## [1] "species_id"
```

Exercise 2

1. Find the column name that is shared between the `plots` table and the `surveys` table. Use that column name for the next question.

Doing it visually, with the colnames function

```
colnames(plots)
```

```
## [1] "plot_id"   "plot_type"
```

```
colnames(surveys)
```

```
## [1] "record_id"       "month"           "day"             "year"
## [5] "plot_id"         "species_id"      "sex"             "hindfoot_length"
## [9] "weight"
```

Automatically with the function `intersect()`

```
intersect(colnames(surveys), colnames(plots))
```

```
## [1] "plot_id"
```

Do the following using a single pipe of code (no nested code nor intermediate variables): Use function inner_join() and filter() to get a data frame with the information from the `surveys` and `plots` tables where the "plot_type" is "Rodent Exclosure".

```
inner_join(surveys, plots, by = "plot_id") %>%
  filter(plot_type == "Rodent Exclosure") %>%
  str()
```

```
## 'data.frame':    4744 obs. of  10 variables:
##  $ record_id      : int  4 11 12 30 32 36 41 55 61 64 ...
##  $ month          : int  7 7 7 7 7 7 7 7 7 8 ...
##  $ day            : int  16 16 16 17 17 17 18 18 18 19 ...
##  $ year           : int  1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 ...
##  $ plot_id        : int  7 5 7 10 10 16 23 23 23 7 ...
##  $ species_id     : chr  "DM" "DS" "DM" "DS" ...
##  $ sex            : chr  "M" "F" "M" "F" ...
##  $ hindfoot_length: int  36 53 38 52 35 22 34 36 35 37 ...
##  $ weight         : int  NA NA NA NA NA NA NA NA NA 48 ...
##  $ plot_type      : chr  "Rodent Exclosure" "Rodent Exclosure" "Rodent Exclosure" "Rodent Exclosure"
```

Other join functions

`left_join()` retains all values from the first table, drops unmatching rows from second

`right_join` drops values from the first table and retaining all values from second

`full_join` keeps all values from both tables

Joining multiple data tables

Can we use the `_join()` function on 3 or more tables at the same time?

```
inner_join(surveys, species, plots)
```

```
## Error in `inner_join()`:
## ! `by` must be a (named) character vector, list, `join_by()` result, or
##   NULL, not a <data.frame> object.
```

No. It does not recognize more than two tables at a time

So we use a pipe an call the join function two or more times (as needed):

```
inner_join(surveys, species, by = "species_id") %>%
  inner_join(plots, by = "plot_id") %>%
  str()
```

```
## 'data.frame':    34786 obs. of  13 variables:
##  $ record_id      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ month          : int  7 7 7 7 7 7 7 7 7 7 ...
##  $ day            : int  16 16 16 16 16 16 16 16 16 16 ...
##  $ year           : int  1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 ...
##  $ plot_id        : int  2 3 2 7 3 1 2 1 1 6 ...
##  $ species_id     : chr  "NL" "NL" "DM" "DM" ...
##  $ sex            : chr  "M" "M" "F" "M" ...
##  $ hindfoot_length: int  32 33 37 36 35 14 NA 37 34 20 ...
##  $ weight         : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ genus          : chr  "Neotoma" "Neotoma" "Dipodomys" "Dipodomys" ...
##  $ species        : chr  "albigula" "albigula" "merriami" "merriami" ...
##  $ taxa           : chr  "Rodent" "Rodent" "Rodent" "Rodent" ...
##  $ plot_type      : chr  "Control" "Long-term Krat Exclosure" "Control" "Rodent Exclosure" ...
```

Exercise 3

1. We want to do an analysis comparing the size of individuals on the "Control" plots to the "Long-term Krat Exclosures".
2. Create a data frame with the "year", "genus", "species", "weight" and "plot_type" for all cases where the plot type is either "Control" or "Long-term Krat Exclosure". Pay attention to typos in lower case and upper case values.
3. Only include cases where the column "taxa" is "Rodent". Remove any records where the "weight" is missing.

```
inner_join(surveys, species, by = "species_id") %>%
  inner_join(plots, by = "plot_id") %>%
  filter(plot_type == "Long term Krat Exclosure" | plot_type == "Control") %>%
  filter(taxa == "Rodent") %>%
  filter(!is.na(weight)) %>%
  select(year, genus, species, weight, plot_type) %>%
  str()
```

```
## 'data.frame':    14652 obs. of  5 variables:
##  $ year     : int  1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 ...
##  $ genus    : chr  "Dipodomys" "Dipodomys" "Dipodomys" "Perognathus" ...
##  $ species  : chr  "merriami" "merriami" "ordii" "flavus" ...
##  $ weight   : int  29 46 52 8 7 22 8 41 15 41 ...
##  $ plot_type: chr  "Control" "Control" "Control" "Control" ...
```

```
# na.rm =  is an argument of functions like mean:
mean(c(NA, 1, 5, 7), na.rm = TRUE)
```

```
## [1] 4.333333
```