Joining Data Tables

Anita Dhillon

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 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")
#View(plots)
#View(surveys)
```

Why do we need to combine or join data tables

How do we join data tables in R

there is a group of functions ’_join()’ that allows us to combine two data tables using values on a shared column

There has to be a shared column; we need three main arguments to run these functions, two data tbles and one column name

The different functions allows us to combine in different ways .

’ inner_join’

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

pipe code

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

How can we explore our combined/joined table

we want to see the difference between the two input tables and the resulting table To see the difference in columns we use 'heads()':

```
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
```

```
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(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_value vs surveys?

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

Exercise 1

```
surveys %>%
inner_join(plots, by = "plot_id") -> joined_table1
  filter(joined_table1, 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
```

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

Automate joining tables/other things with ' intersect'()

```
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"
```

Doing it visually

```
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"
```

Exercise 2

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

```
##   record_id month day year plot_id species_id sex hindfoot_length weight
## 1         4     7  16 1977       7         DM   M              36     NA
## 2        11     7  16 1977       5         DS   F              53     NA
## 3        12     7  16 1977       7         DM   M              38     NA
## 4        30     7  17 1977      10         DS   F              52     NA
## 5        32     7  17 1977      10         DM   F              35     NA
## 6        36     7  17 1977      16         OT   F              22     NA
##          plot_type
## 1 Rodent Exclosure
## 2 Rodent Exclosure
## 3 Rodent Exclosure
## 4 Rodent Exclosure
## 5 Rodent Exclosure
## 6 Rodent Exclosure
```

Joining multiple data tables

can we use the ’_join()’ function on 3 or more tables at the same time (NO)

```
combined <- inner_join(surveys, species, by = "species_id")
combined_final <- inner_join(combined, plots, by = "plot_id")
```

So we can use a Pipe to the join function two or more times (as needed)

```
combined <- surveys |>
inner_join(species, by = "species_id") |>
inner_join(plots, by = "plot_id")
str(combined)
```

```
## ’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" ...
```

other join Functions

‘left_join()’ retains all values from the first table, drops unmatching from second ‘right_join()’ drops values from the first table and retaining all values from second

‘full_join()’ keeps all values from both tables

Exercise 3

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

```
## 'data.frame':    19344 obs. of  5 variables:
##  $ year     : int  1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 ...
##  $ genus    : chr  "Dipodomys" "Dipodomys" "Dipodomys" "Dipodomys" ...
##  $ species  : chr  "merriami" "merriami" "merriami" "ordii" ...
##  $ weight   : int  40 29 46 52 8 22 7 22 8 41 ...
##  $ plot_type: chr  "Long-term Krat Exclosure" "Control" "Control" "Control" ...
```

HOMEWORK (3/13/2023)

##Exercise 4 Create a data frame with the average “hindfoot_length” - I should use the mean function and apply it to hindfoot length for each “species_id_ in each”year” with no null values

```
surveys %>%
  filter(is.na(hindfoot_length)) %>%
group_by(species_id , year) %>%
  summarize(mean(hindfoot_length, na.rm = TRUE))
```

```
## ‘summarise()‘ has grouped output by 'species_id'. You can override using the
## ‘.groups‘ argument.

## # A tibble: 422 x 3
## # Groups:   species_id [45]
##    species_id  year ‘mean(hindfoot_length, na.rm = TRUE)‘
##    <chr>      <int>                                 <dbl>
##  1 ""          1977                                   NaN
##  2 ""          1978                                   NaN
##  3 ""          1979                                   NaN
##  4 ""          1980                                   NaN
##  5 ""          1981                                   NaN
##  6 ""          1982                                   NaN
##  7 ""          1983                                   NaN
##  8 ""          1984                                   NaN
##  9 ""          1985                                   NaN
## 10 ""          1986                                   NaN
## # ... with 412 more rows
```

##Exercise 5

```
inner_join(surveys, species, by = "species_id") %>%
  inner_join(plots, by = "plot_id") %>%
  select(month, day, year, species_id, weight, hindfoot_length) %>%
  filter(is.na(weight)) %>%
  arrange(species_id,desc(weight)) %>%
  head()
```

```
##   month day year species_id weight hindfoot_length
## 1     7  21 1980         AB     NA              NA
## 2     7  21 1980         AB     NA              NA
## 3     7  21 1980         AB     NA              NA
## 4     7  21 1980         AB     NA              NA
## 5    12  15 1980         AB     NA              NA
## 6     1  11 1981         AB     NA              NA
```