Joining data with R

Jonathan de Bruin
14 May 2018

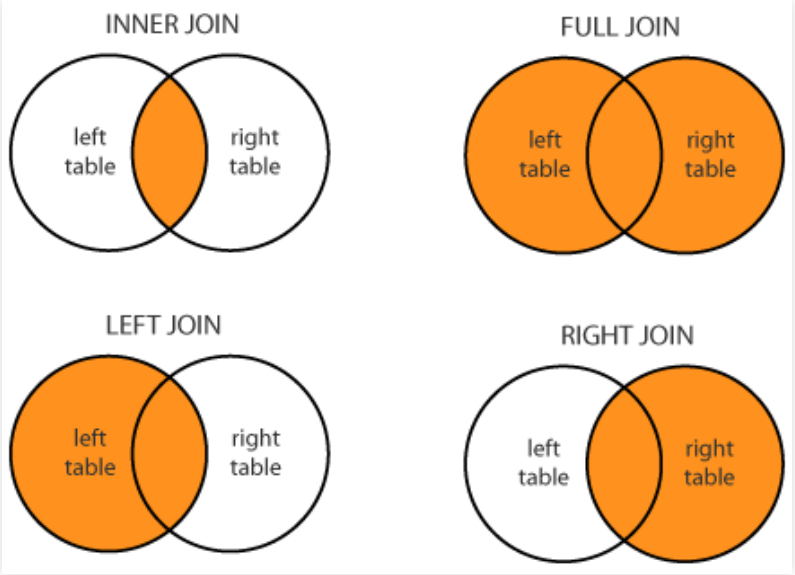
Joining data

- Merge two or more datasets into one
- 'Joining' and 'merging' data mean the same thing
- R implementations are inspired by the SQL database syntax.

Outline

- Simple joins
  - Inner, left, right and outer joins
- Special joins
  - Semi- and anti-joins
- Advanced joins
  - Overlap and fuzzy joins

Often visualized with Venn diagrams



Source: **http://www.dofactory.com/sql/join**

Demo data

```
patient
```

```
  patient_id      name gender        dob
1  189234988     Smith   male 1990-10-10
2  278923732     Jones female 1930-01-09
3  849058094 Zabrinsky female 1953-04-08
```

```
measures
```

```
  patient_id      type value
1  189234988 measure_x  0.17
2  189234988 measure_y 69.00
3  849058094 measure_x  0.25
4  849058094 measure_z  0.33
5  829305840 measure_y 71.00
6  198234988 measure_z  0.34
```

Base R syntax

- Inner join

```
merge(patient, measures, by = 'patient_id')
```

```
  patient_id      name gender        dob      type value
1  189234988     Smith   male 1990-10-10 measure_x  0.17
2  189234988     Smith   male 1990-10-10 measure_y 69.00
3  849058094 Zabrinsky female 1953-04-08 measure_x  0.25
4  849058094 Zabrinsky female 1953-04-08 measure_z  0.33
```

Base R syntax

- Left join

```
merge(patient, measures, by = 'patient_id', all.x = T, all.y = F)
```

```
  patient_id      name gender        dob      type value
1  189234988     Smith   male 1990-10-10 measure_x  0.17
2  189234988     Smith   male 1990-10-10 measure_y 69.00
3  278923732     Jones female 1930-01-09      <NA>    NA
4  849058094 Zabrinsky female 1953-04-08 measure_x  0.25
5  849058094 Zabrinsky female 1953-04-08 measure_z  0.33
```

Base R syntax

- Outer/full join

```
merge(patient, measures, by = 'patient_id', all = T)
```

```
  patient_id      name gender        dob      type value
1  189234988     Smith   male 1990-10-10 measure_x  0.17
2  189234988     Smith   male 1990-10-10 measure_y 69.00
3  198234988      <NA>   <NA>       <NA> measure_z  0.34
4  278923732     Jones female 1930-01-09      <NA>    NA
5  829305840      <NA>   <NA>       <NA> measure_y 71.00
6  849058094 Zabrinsky female 1953-04-08 measure_x  0.25
7  849058094 Zabrinsky female 1953-04-08 measure_z  0.33
```

dplyr R syntax

Import the `dplyr` library

```
library(dplyr)
```

- Inner join

```
inner_join(patient, measures, by = 'patient_id')
```

```
  patient_id      name gender        dob      type value
1  189234988     Smith   male 1990-10-10 measure_x  0.17
2  189234988     Smith   male 1990-10-10 measure_y 69.00
3  849058094 Zabrinsky female 1953-04-08 measure_x  0.25
4  849058094 Zabrinsky female 1953-04-08 measure_z  0.33
```

dplyr R syntax

- Left join

```
left_join(patient, measures, by = 'patient_id')
```

```
  patient_id      name gender        dob      type value
1  189234988     Smith   male 1990-10-10 measure_x  0.17
2  189234988     Smith   male 1990-10-10 measure_y 69.00
3  278923732     Jones female 1930-01-09      <NA>    NA
4  849058094 Zabrinsky female 1953-04-08 measure_x  0.25
5  849058094 Zabrinsky female 1953-04-08 measure_z  0.33
```

dplyr R syntax

- Outer/full join

```
full_join(patient, measures, by = 'patient_id')
```

```
  patient_id      name gender        dob      type value
1  189234988     Smith   male 1990-10-10 measure_x  0.17
2  189234988     Smith   male 1990-10-10 measure_y 69.00
3  278923732     Jones female 1930-01-09      <NA>    NA
4  849058094 Zabrinsky female 1953-04-08 measure_x  0.25
5  849058094 Zabrinsky female 1953-04-08 measure_z  0.33
6  829305840      <NA>   <NA>       <NA> measure_y 71.00
7  198234988      <NA>   <NA>       <NA> measure_z  0.34
```

Semi-join

- Help: *'Returns all rows from `x` where there are matching values in `y`, keeping just columns from `x`.'*
- **All patients with measures!**

dplyr code:

```
semi_join(patient, measures, by = 'patient_id')
```

```
  patient_id      name gender        dob
1  189234988     Smith   male 1990-10-10
2  849058094 Zabrinsky female 1953-04-08
```

Semi-join

- Help: *'Returns all rows from `x` where there are matching values in `y`, keeping just columns from `x`.'*

- **All patients with measures!**

More complex syntax in base R:

```
subset(patient, patient$patient_id %in% measures$patient_id)
```

```
  patient_id      name gender        dob
1  189234988     Smith   male 1990-10-10
3  849058094 Zabrinsky female 1953-04-08
```

Anti-join

- Help: *'Returns all rows from x where there are not matching values in y, keeping just columns from x.'*
- **All patients without measures!**

dplyr code:

```
anti_join(patient, measures, by = 'patient_id')
```

```
  patient_id  name gender        dob
1  278923732 Jones female 1930-01-09
```

Anti-join

- Help: *'Returns all rows from x where there are not matching values in y, keeping just columns from x.'*
- **All patients without measures!**

More complex syntax in base R:

```
subset(patient, !(patient$patient_id %in% measures$patient_id))
```

```
  patient_id  name gender        dob
2  278923732 Jones female 1930-01-09
```

Fuzzy (inner) join

- Join datasets on partially matching keys
- People make typos (insertions, deletions and substitutions)

Not implement in base R, but possible to write by yourself

```
library(reshape2)

distance_matrix <- adist(patient$patient_id, measures$patient_id)

distance_matrix_tidy <- melt(distance_matrix)
colnames(distance_matrix_tidy) <- c(
    "record_pat", "record_meas", "distance"
)

# consider all records with less than 5 mistakes are matches
matches <- distance_matrix_tidy[distance_matrix_tidy$distance < 3,]
matches$distance <- NULL

df_fuzzy_joined <- merge(
    merge(matches, patient, by.x='record_pat', by.y=0),
    measures, by.x='record_meas', by.y=0
)

df_fuzzy_joined
```

```
  record_meas record_pat patient_id.x      name gender        dob
1           1          1    189234988     Smith   male 1990-10-10
```

```
2           2          1    189234988     Smith   male 1990-10-10
3           3          3    849058094 Zabrinsky female 1953-04-08
4           4          3    849058094 Zabrinsky female 1953-04-08
5           6          1    189234988     Smith   male 1990-10-10
  patient_id.y      type value
1    189234988 measure_x  0.17
2    189234988 measure_y 69.00
3    849058094 measure_x  0.25
4    849058094 measure_z  0.33
5    198234988 measure_z  0.34
```

Fuzzy (inner) join (step 1)

- Compute the string similarity matrix.

```
distance_matrix <- adist(patient$patient_id, measures$patient_id)

distance_matrix
```

```
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    0    0    8    8    7    2
[2,]    6    6    9    9    9    7
[3,]    8    8    0    0    5    8
```

Fuzzy (inner) join (step 2)

- Make the data tidy.

```
library(reshape2)

distance_matrix_tidy <- melt(distance_matrix)
colnames(distance_matrix_tidy) <- c(
    "record_pat", "record_meas", "distance"
)

distance_matrix_tidy
```

```
   record_pat record_meas distance
1           1           1        0
2           2           1        6
3           3           1        8
4           1           2        0
5           2           2        6
6           3           2        8
7           1           3        8
8           2           3        9
9           3           3        0
10          1           4        8
11          2           4        9
12          3           4        0
13          1           5        7
14          2           5        9
15          3           5        5
16          1           6        2
17          2           6        7
18          3           6        8
```

Fuzzy (inner) join (step 3)

- Set the maximum number of mistakes to less than 3.

```
# consider all records with less than 5 mistakes are matches
matches <- distance_matrix_tidy[distance_matrix_tidy$distance < 3,]
matches$distance <- NULL

matches
```

```
   record_pat record_meas
1           1           1
4           1           2
9           3           3
12          3           4
16          1           6
```

Fuzzy (inner) join (step 4)

- Perform a double join.
- 189234988 and 198234988 match!!

```
df_fuzzy_joined <- merge(
    merge(matches, patient, by.x='record_pat', by.y=0),
    measures, by.x='record_meas', by.y=0
)

df_fuzzy_joined
```

```
  record_meas record_pat patient_id.x      name gender        dob
1           1          1    189234988     Smith   male 1990-10-10
2           2          1    189234988     Smith   male 1990-10-10
3           3          3    849058094 Zabrinsky female 1953-04-08
4           4          3    849058094 Zabrinsky female 1953-04-08
5           6          1    189234988     Smith   male 1990-10-10
  patient_id.y      type value
1    189234988 measure_x  0.17
2    189234988 measure_y 69.00
3    849058094 measure_x  0.25
4    849058094 measure_z  0.33
5    198234988 measure_z  0.34
```