Cheatsheet for dplyr join functions

Jenny Bryan

11 September, 2014

Why the cheatsheet

Examples for those of us who don't speak SQL so good. There are lots of Venn diagrams re: SQL joins on the interwebs, but I wanted R examples.

Full documentation for the `dplyr` package, which is developed by Hadley Wickham and Romain Francois on GitHub.

Working with two small data.frames, `superheroes` and `publishers`.

```
suppressPackageStartupMessages(library(dplyr))

superheroes <-
  c("    name, alignment, gender,         publisher",
    " Magneto,       bad,   male,            Marvel",
    "   Storm,      good, female,            Marvel",
    "Mystique,       bad, female,            Marvel",
    "  Batman,      good,   male,                DC",
    "   Joker,       bad,   male,                DC",
    "Catwoman,       bad, female,                DC",
    " Hellboy,      good,   male, Dark Horse Comics")
superheroes <- read.csv(text = superheroes, strip.white = TRUE)

publishers <-
  c("publisher, yr_founded",
    "       DC,       1934",
    "   Marvel,       1939",
    "    Image,       1992")
publishers <- read.csv(text = publishers, strip.white = TRUE)
```

Sorry, cheat sheet does not illustrate “multiple match” situations terribly well.

Sub-plot: watch the row and variable order of the join results for a healthy reminder of why it’s dangerous to rely on any of that in an analysis.

inner_join(superheroes, publishers)

> inner_join(x, y): return all rows from x where there are matching values in y, and all columns from x and y. If there are multiple matches between x and y, all combination of the matches are returned

```
(ijsp <- inner_join(superheroes, publishers))
```

```
## Joining by: "publisher"
```

```
##       name alignment gender publisher yr_founded
## 1  Magneto       bad   male    Marvel       1939
## 2    Storm      good female    Marvel       1939
## 3 Mystique       bad female    Marvel       1939
## 4   Batman      good   male        DC       1934
## 5    Joker       bad   male        DC       1934
## 6 Catwoman       bad female        DC       1934
```

We lose Hellboy in the join because, although he appears in `x = superheroes`, his publisher Dark Horse Comics does not appear in `y = publishers`. The join result has all variables from `x = superheroes` plus `yr_founded`, from `y`.

superheroes

name	alignment	gender	publisher
Magneto	bad	male	Marvel
Storm	good	female	Marvel
Mystique	bad	female	Marvel
Batman	good	male	DC
Joker	bad	male	DC
Catwoman	bad	female	DC
Hellboy	good	male	Dark Horse Comics

publishers

publisher	yr_founded
DC	1934
Marvel	1939
Image	1992

inner_join(x = superheroes, y = publishers)

name	alignment	gender	publisher	yr_founded
Magneto	bad	male	Marvel	1939
Storm	good	female	Marvel	1939
Mystique	bad	female	Marvel	1939
Batman	good	male	DC	1934
Joker	bad	male	DC	1934
Catwoman	bad	female	DC	1934

semi_join(superheroes, publishers)

> semi_join(x, y): return all rows from x where there are matching values in y, keeping just columns from x. A semi join differs from an inner join because an inner join will return one row of x for each matching row of y, where a semi join will never duplicate rows of x.

```
(sjsp <- semi_join(superheroes, publishers))
```

```
## Joining by: "publisher"
```

```
##       name alignment gender publisher
## 1   Batman      good   male        DC
## 2    Joker       bad   male        DC
## 3 Catwoman       bad female        DC
## 4  Magneto       bad   male    Marvel
## 5    Storm      good female    Marvel
## 6 Mystique       bad female    Marvel
```

We get a similar result as with `inner_join()` but the join result contains only the variables originally found in `x = superheroes`.

superheroes

name	alignment	gender	publisher
Magneto	bad	male	Marvel
Storm	good	female	Marvel
Mystique	bad	female	Marvel
Batman	good	male	DC
Joker	bad	male	DC
Catwoman	bad	female	DC
Hellboy	good	male	Dark Horse Comics

publishers

publisher	yr_founded
DC	1934
Marvel	1939
Image	1992

semi-join(x = superheroes, y = publishers)

name	alignment	gender	publisher
Batman	good	male	DC
Joker	bad	male	DC
Catwoman	bad	female	DC
Magneto	bad	male	Marvel
Storm	good	female	Marvel
Mystique	bad	female	Marvel

left_join(superheroes, publishers)

left_join(x, y): return all rows from x, and all columns from x and y. If there are multiple matches between x and y, all combination of the matches are returned

```
(ljsp <- left_join(superheroes, publishers))
```

```
## Joining by: "publisher"
```

```
##       name alignment gender         publisher yr_founded
## 1  Magneto       bad   male            Marvel       1939
## 2    Storm      good female            Marvel       1939
## 3 Mystique       bad female            Marvel       1939
## 4   Batman      good   male                DC       1934
## 5    Joker       bad   male                DC       1934
## 6 Catwoman       bad female                DC       1934
## 7  Hellboy      good   male Dark Horse Comics         NA
```

We basically get `x = superheroes` back, but with the addition of variable `yr_founded`, which is unique to `y = publishers`. Hellboy, whose publisher does not appear in `y = publishers`, has an `NA` for `yr_founded`.

superheroes

name	alignment	gender	publisher
Magneto	bad	male	Marvel
Storm	good	female	Marvel
Mystique	bad	female	Marvel
Batman	good	male	DC
Joker	bad	male	DC
Catwoman	bad	female	DC
Hellboy	good	male	Dark Horse Comics

publishers

publisher	yr_founded
DC	1934
Marvel	1939
Image	1992

left_join(x = superheroes, y = publishers)

name	alignment	gender	publisher	yr_founded
Magneto	bad	male	Marvel	1939
Storm	good	female	Marvel	1939
Mystique	bad	female	Marvel	1939
Batman	good	male	DC	1934
Joker	bad	male	DC	1934
Catwoman	bad	female	DC	1934
Hellboy	good	male	Dark Horse Comics	NA

anti_join(superheroes, publishers)

anti_join(x, y): return all rows from x where there are not matching values in y, keeping just columns from x

```
(ajsp <- anti_join(superheroes, publishers))
```

```
## Joining by: "publisher"
```

```
##      name alignment gender         publisher
## 1 Hellboy      good   male Dark Horse Comics
```

We keep **only** Hellboy now (and do not get `yr_founded`).

superheroes

name	alignment	gender	publisher
Magneto	bad	male	Marvel
Storm	good	female	Marvel
Mystique	bad	female	Marvel
Batman	good	male	DC
Joker	bad	male	DC
Catwoman	bad	female	DC
Hellboy	good	male	Dark Horse Comics

publishers

publisher	yr_founded
DC	1934
Marvel	1939
Image	1992

anti_join(x = superheroes, y = publishers)

name	alignment	gender	publisher
Hellboy	good	male	Dark Horse Comics

inner_join(publishers, superheroes)

inner_join(x, y): return all rows from x where there are matching values in y, and all columns from x and y. If there are multiple matches between x and y, all combination of the matches are returned

```
(ijps <- inner_join(publishers, superheroes))
```

```
## Joining by: "publisher"
```

```
##   publisher yr_founded     name alignment gender
## 1    Marvel       1939  Magneto       bad   male
## 2    Marvel       1939    Storm      good female
## 3    Marvel       1939 Mystique       bad female
## 4        DC       1934   Batman      good   male
## 5        DC       1934    Joker       bad   male
## 6        DC       1934 Catwoman       bad female
```

In a way, this does illustrate multiple matches, if you think about it from the `x = publishers` direction. Every publisher that has a match in `y = superheroes` appears multiple times in the result, once for each match. In fact, we're getting the same result as with `inner_join(superheroes, publishers)`, up to variable order (which you should also never rely on in an analysis).

superheroes

publisher	yr_founded
DC	1934
Marvel	1939
Image	1992

publishers

name	alignment	gender	publisher
Magneto	bad	male	Marvel
Storm	good	female	Marvel
Mystique	bad	female	Marvel
Batman	good	male	DC
Joker	bad	male	DC
Catwoman	bad	female	DC
Hellboy	good	male	Dark Horse Comics

inner_join(x = publishers, y = superheroes)

publisher	yr_founded	name	alignment	gender
Marvel	1939	Magneto	bad	male
Marvel	1939	Storm	good	female
Marvel	1939	Mystique	bad	female
DC	1934	Batman	good	male
DC	1934	Joker	bad	male
DC	1934	Catwoman	bad	female

semi_join(publishers, superheroes)

semi_join(x, y): return all rows from x where there are matching values in y, keeping just columns from x. A semi join differs from an inner join because an inner join will return one row of x for each matching row of y, where a semi join will never duplicate rows of x.

```
(sjps <- semi_join(x = publishers, y = superheroes))
```

```
## Joining by: "publisher"

##   publisher yr_founded
## 1    Marvel        1939
## 2        DC        1934
```

Now the effects of switching the `x` and `y` roles is more clear. The result resembles `x = publishers`, but the publisher Image is lost, because there are no observations where `publisher == "Image"` in `y = superheroes`.

superheroes

publisher	yr_founded
DC	1934
Marvel	1939
Image	1992

publishers

name	alignment	gender	publisher
Magneto	bad	male	Marvel
Storm	good	female	Marvel
Mystique	bad	female	Marvel
Batman	good	male	DC
Joker	bad	male	DC
Catwoman	bad	female	DC
Hellboy	good	male	Dark Horse Comics

semi-join(x = publishers, y = superheroes)

publisher	yr_founded
Marvel	1939
DC	1934

left_join(publishers, superheroes)

> left_join(x, y): return all rows from x, and all columns from x and y. If there are multiple matches between x and y, all combination of the matches are returned

```
(ljps <- left_join(publishers, superheroes))

## Joining by: "publisher"

##   publisher yr_founded     name alignment gender
## 1        DC        1934   Batman      good   male
## 2        DC        1934    Joker       bad   male
## 3        DC        1934 Catwoman       bad female
## 4    Marvel        1939  Magneto       bad   male
## 5    Marvel        1939    Storm      good female
## 6    Marvel        1939 Mystique       bad female
## 7     Image        1992     <NA>      <NA>   <NA>
```

We get a similar result as with `inner_join()` but the publisher Image survives in the join, even though no superheroes from Image appear in `y = superheroes`. As a result, Image has `NA`s for `name`, `alignment`, and `gender`.

publishers

publisher	yr_founded
DC	1934
Marvel	1939
Image	1992

superheroes

name	alignment	gender	publisher
Magneto	bad	male	Marvel
Storm	good	female	Marvel
Mystique	bad	female	Marvel
Batman	good	male	DC
Joker	bad	male	DC
Catwoman	bad	female	DC
Hellboy	good	male	Dark Horse Comics

left_join(x = publishers, y = superheroes)

publisher	yr_founded	name	alignment	gender
DC	1934	Batman	good	male
DC	1934	Joker	bad	male
DC	1934	Catwoman	bad	female
Marvel	1939	Magneto	bad	male
Marvel	1939	Storm	good	female
Marvel	1939	Mystique	bad	female
Image	1992	NA	NA	NA

anti_join(publishers, superheroes)

anti_join(x, y): return all rows from x where there are not matching values in y, keeping just columns from x

```
(ajps <- anti_join(publishers, superheroes))
```

```
## Joining by: "publisher"
```

```
##   publisher yr_founded
## 1     Image       1992
```

We keep **only** publisher Image now (and the variables found in `x = publishers`).

publishers

publisher	yr_founded
DC	1934
Marvel	1939
Image	1992

superheroes

name	alignment	gender	publisher
Magneto	bad	male	Marvel
Storm	good	female	Marvel
Mystique	bad	female	Marvel
Batman	good	male	DC
Joker	bad	male	DC
Catwoman	bad	female	DC
Hellboy	good	male	Dark Horse Comics

anti_join(x = publishers, y = superheroes)

publisher	yr_founded
Image	1992

NOT dplyr: merge(superheroes, publishers, all = TRUE)

What if you want to merge two data.frames and keep rows that appear in *either*? In SQL jargon, this is an outer join and is not yet implemented in `dplyr`, though it will come. In the meantime, you could use `merge()` from base R.

> merge(x, y): Merge two data frames by common columns or row names, or do other versions of database join operations

```
(OJsp <- merge(superheroes, publishers, all = TRUE))
```

```
##           publisher     name alignment gender yr_founded
## 1 Dark Horse Comics  Hellboy      good   male         NA
## 2                DC   Batman      good   male       1934
## 3                DC    Joker       bad   male       1934
## 4                DC Catwoman       bad female       1934
## 5            Marvel  Magneto       bad   male       1939
## 6            Marvel    Storm      good female       1939
## 7            Marvel Mystique       bad female       1939
## 8             Image     <NA>      <NA>   <NA>       1992
```

We keep Hellboy (whose publisher Dark Horse Comics is not in `publishers`) and Image (a publisher with no superheroes in `superheroes`) and get variables from both data.frames. Therefore observations for which there is no match in the two data.frames carry `NA`s in the variables from the other data source.

superheroes

name	alignment	gender	publisher
Magneto	bad	male	Marvel
Storm	good	female	Marvel
Mystique	bad	female	Marvel
Batman	good	male	DC
Joker	bad	male	DC
Catwoman	bad	female	DC
Hellboy	good	male	Dark Horse

publishers

publisher	yr_founded
DC	1934
Marvel	1939
Image	1992

merge(superheroes, publishers, all = TRUE)

publisher	name	alignment	gender	yr_founded
Dark Horse Comics	Hellboy	good	male	NA
DC	Batman	good	male	1934
DC	Joker	bad	male	1934
DC	Catwoman	bad	female	1934
Marvel	Magneto	bad	male	1939
Marvel	Storm	good	female	1939

Comics		Marvel	Mystique	bad	female	1939
		Image	NA	NA	NA	1992

sessionInfo()

```
sessionInfo()
```

```
## R version 3.1.0 (2014-04-10)
## Platform: x86_64-apple-darwin10.8.0 (64-bit)
##
## locale:
## [1] en_CA.UTF-8/en_CA.UTF-8/en_CA.UTF-8/C/en_CA.UTF-8/en_CA.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] dplyr_0.2.0.99
##
## loaded via a namespace (and not attached):
##  [1] assertthat_0.1   digest_0.6.4     evaluate_0.5.5   formatR_0.10
##  [5] htmltools_0.2.4  knitr_1.6        magrittr_1.0.1   parallel_3.1.0
##  [9] Rcpp_0.11.1      rmarkdown_0.2.64 stringr_0.6.2    tools_3.1.0
## [13] yaml_2.1.13
```