Joining data from multiple sources

K Arnold, based on DSBox

Logistics

- Lab 4 due today
- Discussion 2 reply due tomorrow
    - Not just: "you made a good point"
    - Some good examples: Krista, Lauren (replies to Kaitlyn)
    - Final post not officially due till next week, but some done already. Good examples: Advait, Ricky.
- hw3 and hw4 due Wed
- Quiz 5 on Thursday: no more infinite tries.

One month till Election Day! Are you registered to vote?

Discussion 3: Critique a graph you find

Draft proposal:

- *Collect* examples of visualizations that have been used to make a political or social argument (whether or not you agree!)
- *Post a critique* as Discussion 3
- later (as Homework): replicate the visual yourself, write up your response

A note on `mutate`

- Badly named. Think "add_computed_column".
- **DON'T** think of it operating on a variable ("mutate the ride's start time").
- **DO** think of it operating on a data frame ("add a column computed by flooring the start time)

Q&A

> `count` vs `group_by %>% summarize`?

`count(x)` is (mostly) shorthand for `group_by(x) %>% summarize(n = n())`

> `select` vs `filter`?

Maybe should have been named `select_columns` and `select_rows`.

> Where do I get the specific lines of code I need?

Good question *only if* you have a clear idea of what you want to do. Sketch out a very specific example of the output you want.

Q&A

> What's on the tests?

- All assessments are open-everything (except for getting help from other people).
- Quizzes will become 1-attempt and timed soon.
- Midterm and final are both (mainly) projects.

> uh, `lubridate`??

Most confusions I've seen have actually been about data manipulation concepts, not `lubridate` itself.
If the assignment instructions don't give you all the syntax and parameters you need, let me know.

Q&A

> Cohorts?

- If your repo has your own name on it, it's solo. Team name? It's team.

> Merge conflicts?

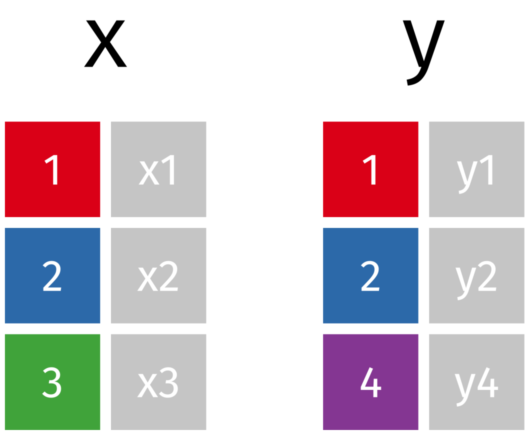
- We'll deliberately make one together this week so you know how to deal with it.

Q&A

> joins?

Joining data frames

- I have a data frame x (e.g., Covid confirmed cases)
- I want extra information about things in x (e.g., population)
- Some other table, y, has that information. "Joins" let me look it up.
- Needs a key : what has to match. Must match *exactly*.



x

1	x1
2	x2
3	x3

y

1	y1
2	y2
4	y4

Graphics thanks to tidyexplain

Types of joins

What to do when things don't exactly line up? Start with all rows, but:

- **full** or **outer**: Leave blanks (`NA`) for mismatches
- **inner**: Drop rows with any mismatches
- **left** / **right**: Drop rows where one of the sides has a mismatch

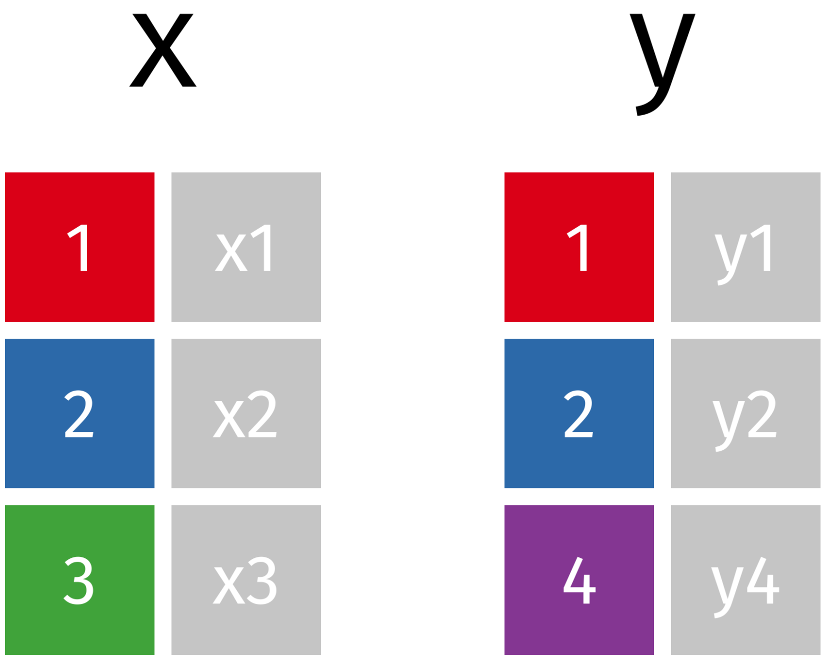
Setup

```
x
```

```
## # A tibble: 3 x 2
##     key xdata
##   <dbl> <chr>
## 1     1 x1
## 2     2 x2
## 3     3 x3
```

```
y
```

```
## # A tibble: 3 x 2
##     key ydata
##   <dbl> <chr>
## 1     1 y1
## 2     2 y2
## 3     4 y4
```



x

1	x1
2	x2
3	x3

y

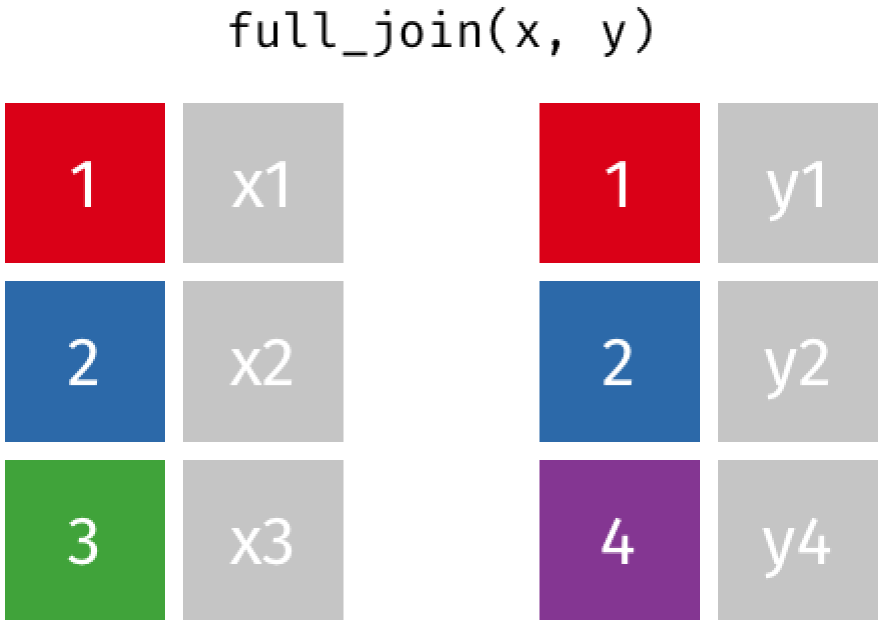
1	y1
2	y2
4	y4

full_join()

All rows from both x and y. Leave NA for mismatches.

```
full_join(x, y, by = "key")
```

```
## # A tibble: 4 x 3
##     key xdata ydata
##   <dbl> <chr> <chr>
## 1     1 x1    y1
## 2     2 x2    y2
## 3     3 x3    <NA>
## 4     4 <NA>  y4
```



full_join(x, y)

1	x1
2	x2
3	x3

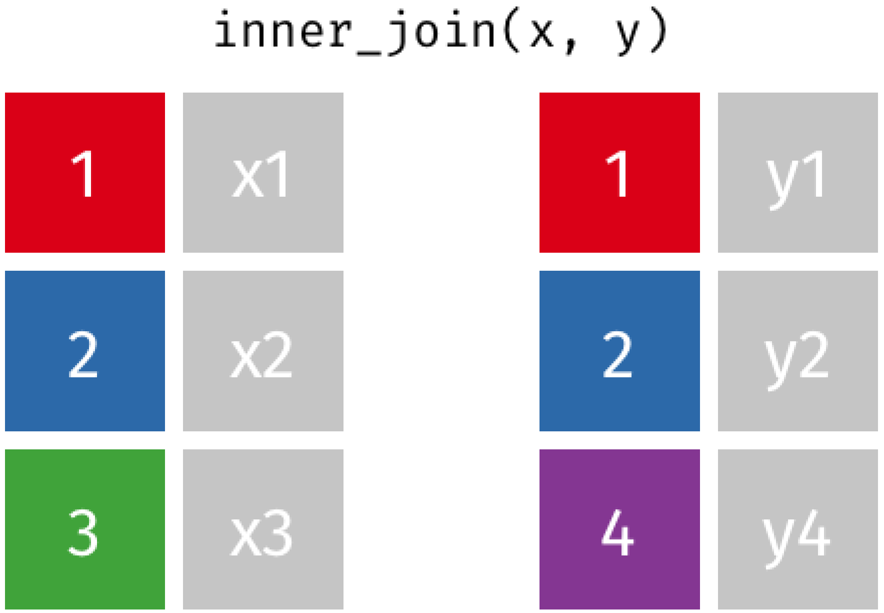
1	y1
2	y2
4	y4

inner_join()

All matching rows. Drops mismatches.

```
inner_join(x, y, by = "key")
```

```
## # A tibble: 2 x 3
##     key xdata ydata
##   <dbl> <chr> <chr>
## 1     1 x1    y1
## 2     2 x2    y2
```

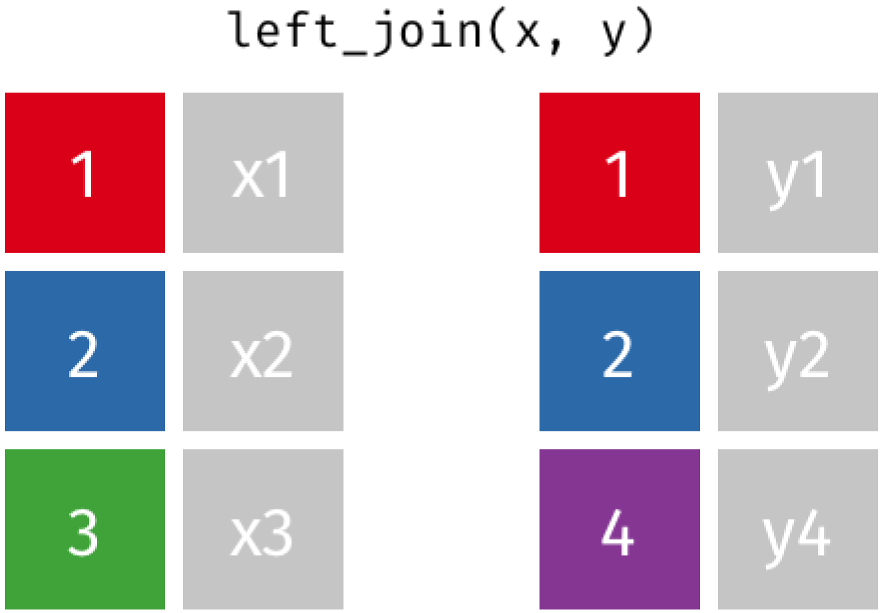


left_join()

All rows from x.

```
left_join(x, y, by = "key")
```

```
## # A tibble: 3 x 3
##     key xdata ydata
##   <dbl> <chr> <chr>
## 1     1 x1    y1
## 2     2 x2    y2
## 3     3 x3    <NA>
```



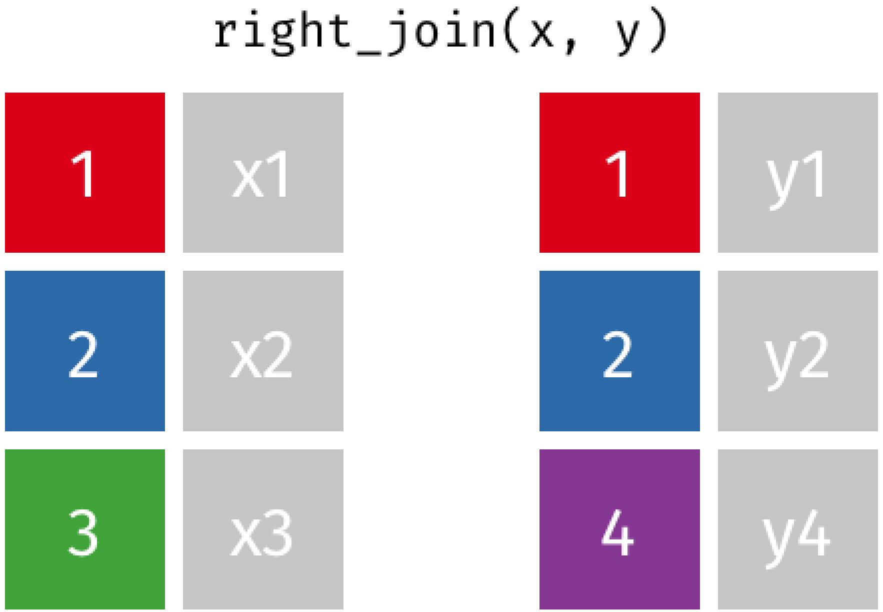
right_join()

All rows from y.

```
right_join(x, y)
```

```
## Joining, by = "key"
```

```
## # A tibble: 3 x 3
##     key xdata ydata
##   <dbl> <chr> <chr>
## 1     1 x1    y1
## 2     2 x2    y2
## 3     4 <NA>  y4
```



Summary

- `full_join()`: all rows from both `x` and `y`
- `inner_join()`: all *matching* rows from `x` where there are matching values in y. Multiple matches? Return all combinations.
- `left_join()`: all rows from x
- `right_join()`: all rows from y

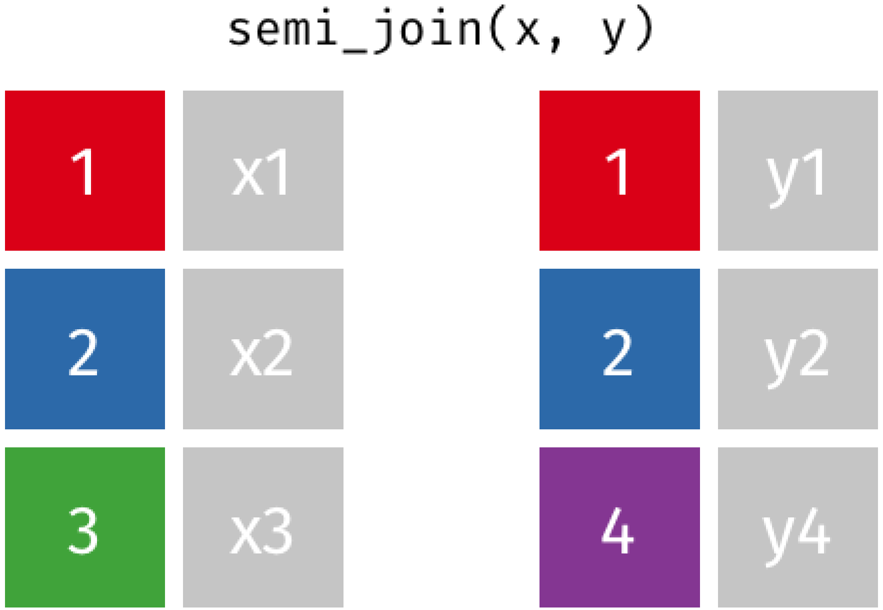
^ are called *mutating joins* (by analogy to the `mutate` verb). Sometimes useful: *filtering* joins and *nest* join:

- `semi_join()`: include a row from `x` only if there's some match in `y`
- `anti_join()`: include a row from `x` only if there's *no* match in `y`
- `nest_join()`: get bundles of all matching rows from `y` (most flexible)

semi_join()

```r
semi_join(x, y, by = "key")
```

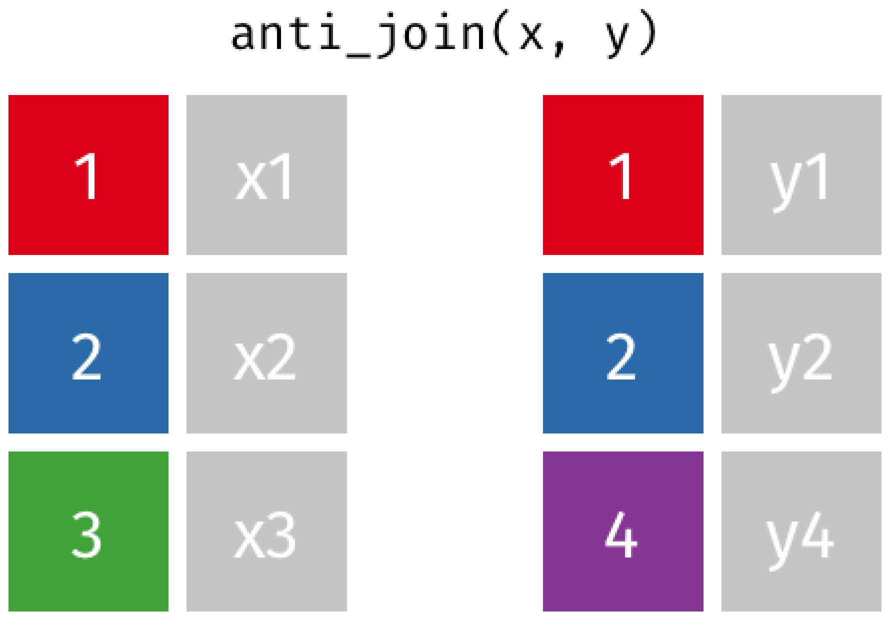
```
## # A tibble: 2 x 2
##     key xdata
##   <dbl> <chr>
## 1     1 x1
## 2     2 x2
```



anti_join()

```
anti_join(x, y, by = "key")
```

```
## # A tibble: 1 x 2
##     key xdata
##   <dbl> <chr>
## 1     3 x3
```



anti_join(x, y)

We want to keep all rows and columns from `confirmed_cases` and add a column for corresponding populations from `population`. Which join function should we use?