Joining data from multiple sources

DATA 202 21FA, based on datasciencebox.org

Q&A

> What does `%in%` do?

- Don't confuse with `%>%` or `==`

```
nobel %>% filter(country == c("USA", "France"))
```

(doesn't work!)

```
nobel %>% filter(country %in% c("USA", "France"))
```

```
# A tibble: 388 × 26
     id firstname surname    year category  affiliation      city
  <dbl> <chr>     <chr>     <dbl> <chr>     <chr>            <chr>
1     4 Henri     Becquerel  1903 Physics   École Polytech… Paris
2     5 Pierre    Curie      1903 Physics   École municipa… Paris
3     6 Marie     Curie      1911 Chemistry Sorbonne Unive… Paris
4    11 Albert A. Michelson  1907 Physics   University of … Chic…
5    12 Gabriel   Lippmann   1908 Physics   Sorbonne Unive… Paris
6    14 Ferdinand Braun      1909 Physics   Strasbourg Uni… Stra…
# … with 382 more rows, and 19 more variables: country <chr>
```

Q&A

> How do I remember all this syntax and keep everything straight?

It takes work. But you can do it. Tips:

- Don't "just make it work".
- Understand *why* something works (explain it!)
- Try out variations!

Data: Women in science

Information on 10 women in science who changed the world

name
Ada Lovelace
Marie Curie
Janaki Ammal
Chien-Shiung Wu
Katherine Johnson
Rosalind Franklin
Vera Rubin
Gladys West
Flossie Wong-Staal
Jennifer Doudna

Source: Discover Magazine

Inputs

professions   dates   notability

```
professions
```

```
# A tibble: 10 × 2
   name                 profession
   <chr>                <chr>
 1 Ada Lovelace         Mathematician
 2 Marie Curie          Physicist and Chemist
 3 Janaki Ammal         Botanist
 4 Chien-Shiung Wu      Physicist
 5 Katherine Johnson    Mathematician
 6 Rosalind Franklin    Chemist
 7 Vera Rubin           Astronomer
 8 Gladys West          Mathematician
 9 Flossie Wong-Staal Virologist and Molecular Biologist
10 Jennifer Doudna      Biochemist
```

Desired output

```
# A tibble: 10 × 5
   name               profession  birth_year death_year known_for
   <chr>              <chr>            <int>      <int> <chr>
 1 Ada Lovelace       Mathematic…         NA         NA first co…
 2 Marie Curie        Physicist …         NA         NA theory o…
 3 Janaki Ammal       Botanist          1897       1984 hybrid s…
 4 Chien-Shiung Wu    Physicist         1912       1997 confim a…
 5 Katherine Johnson  Mathematic…       1918       2020 calculat…
 6 Rosalind Franklin  Chemist           1920       1958 <NA>
 7 Vera Rubin         Astronomer        1928       2016 existenc…
 8 Gladys West        Mathematic…       1930         NA mathemat…
 9 Flossie Wong-Staal Virologist…       1947         NA first sc…
10 Jennifer Doudna    Biochemist        1964         NA one of t…
```

First try: paste them together

name	profession	known_for
Ada Lovelace	Mathematician	first computer algorithm
Marie Curie	Physicist and Chemist	confim and refine theory of radioactive beta decy, Wu experiment overturning theory of parity
Janaki Ammal	Botanist	first scientist to clone HIV and create a map of its genes which led to a test for the virus
Chien-Shiung Wu	Physicist	mathematical modeling of the shape of the Earth which served as the foundation of GPS technology
Katherine Johnson	Mathematician	hybrid species, biodiversity protection
Rosalind Franklin	Chemist	one of the primary developers of CRISPR, a ground-breaking technology for editing genomes
Vera Rubin	Astronomer	calculations of orbital mechanics critical to sending the first Americans into space
Gladys	Mathematician	theory of radioactivity, discovery of elements polonium and radium, first woman to win a Nobel

What was wrong?

What was wrong?

How do we know which rows match up?

What was wrong?

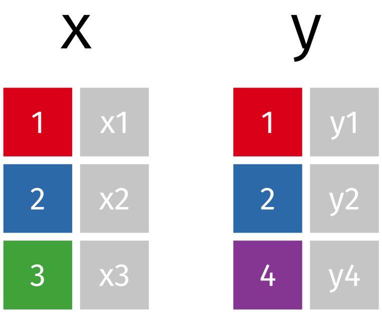
How do we know which rows match up?

Need a **key**.

> What can serve as a **key** for this data?

Mutating joins

- I have a data frame x
- I want extra information about things in x
- Some other table, y, has that information.
- A "join" lets me look it up.



x

1	x1
2	x2
3	x3

y

1	y1
2	y2
4	y4

Graphics thanks to tidyexplain

Types of joins

If x and y match up one-to-one, no difference.

What to do when things don't exactly line up?

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

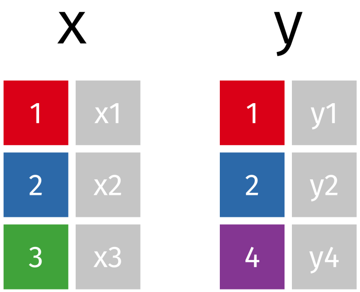
Setup

```
x
```

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

```
y
```

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



x

1	x1
2	x2
3	x3

y

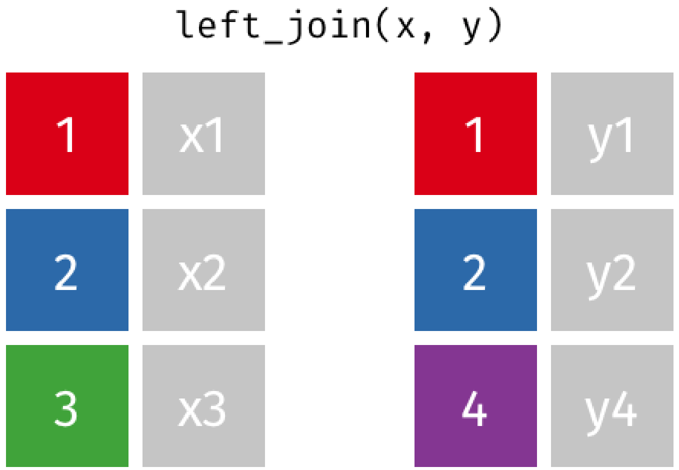
1	y1
2	y2
4	y4

left_join()

All rows from x.

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

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



left_join(x, y)

1	x1
2	x2
3	x3

1	y1
2	y2
4	y4

left_join()

```
professions %>%
  left_join(dates)
```

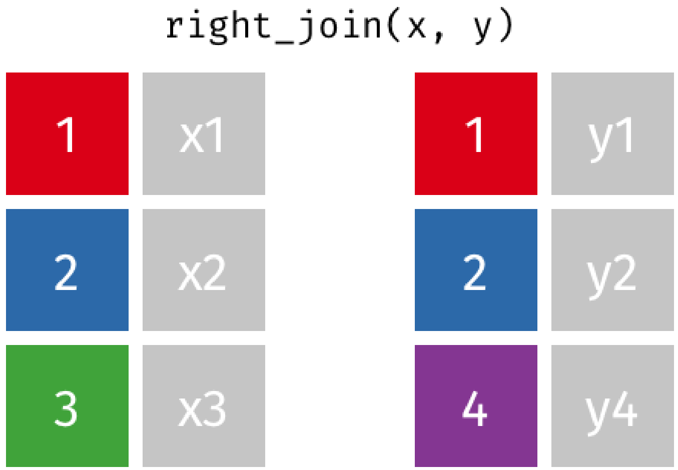
```
# A tibble: 10 × 4
   name               profession                 birth_year death_year
   <chr>              <chr>                           <int>      <int>
 1 Ada Lovelace       Mathematician                      NA         NA
 2 Marie Curie        Physicist and Chemist              NA         NA
 3 Janaki Ammal       Botanist                         1897       1984
 4 Chien-Shiung Wu    Physicist                        1912       1997
 5 Katherine Johnson  Mathematician                    1918       2020
 6 Rosalind Franklin  Chemist                          1920       1958
 7 Vera Rubin         Astronomer                       1928       2016
 8 Gladys West        Mathematician                    1930         NA
 9 Flossie Wong-Staal Virologist and Molec…            1947         NA
10 Jennifer Doudna    Biochemist                       1964         NA
```

right_join()

All rows from y.

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

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


right_join(x, y)

right_join()

```
professions %>%
  right_join(dates)
```

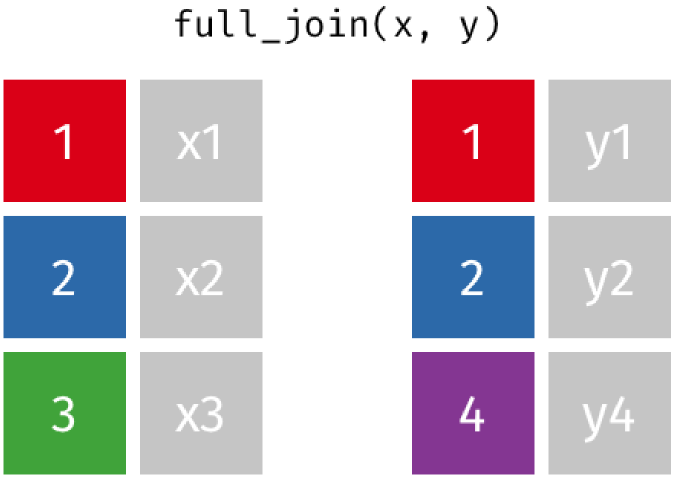
```
# A tibble: 8 × 4
  name                profession                 birth_year death_year
  <chr>               <chr>                           <int>      <int>
1 Janaki Ammal        Botanist                         1897       1984
2 Chien-Shiung Wu     Physicist                        1912       1997
3 Katherine Johnson   Mathematician                    1918       2020
4 Rosalind Franklin   Chemist                          1920       1958
5 Vera Rubin          Astronomer                       1928       2016
6 Gladys West         Mathematician                    1930         NA
7 Flossie Wong-Staal Virologist and Molecu…           1947         NA
8 Jennifer Doudna     Biochemist                       1964         NA
```

full_join()

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

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

```
# A tibble: 4 × 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()

```
dates %>%
  full_join(notability)
```

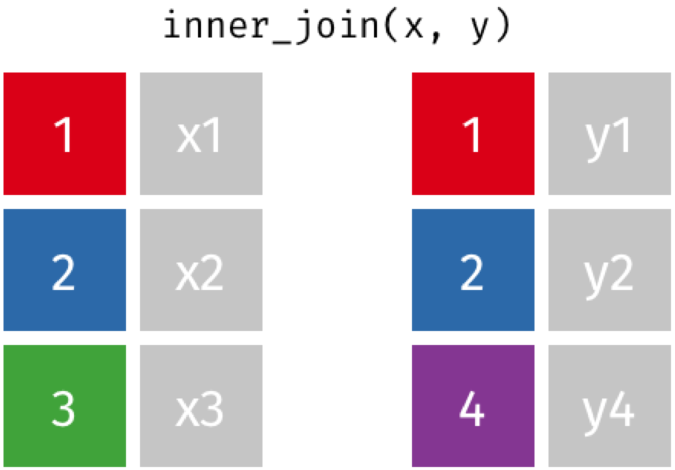
```
# A tibble: 10 × 4
   name               birth_year death_year known_for
   <chr>                   <int>      <int> <chr>
 1 Janaki Ammal             1897       1984 hybrid species, biod…
 2 Chien-Shiung Wu          1912       1997 confim and refine th…
 3 Katherine Johnson        1918       2020 calculations of orbi…
 4 Rosalind Franklin        1920       1958 <NA>
 5 Vera Rubin               1928       2016 existence of dark ma…
 6 Gladys West              1930         NA mathematical modelin…
 7 Flossie Wong-Staal       1947         NA first scientist to c…
 8 Jennifer Doudna          1964         NA one of the primary d…
 9 Ada Lovelace               NA         NA first computer algor…
10 Marie Curie                NA         NA theory of radioactiv…
```

inner_join()

All matching rows. Drops mismatches.

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

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



inner_join()

```
dates %>%
  inner_join(notability)
```

```
# A tibble: 7 × 4
  name                 birth_year death_year known_for
  <chr>                     <int>      <int> <chr>
1 Janaki Ammal               1897       1984 hybrid species, biodi…
2 Chien-Shiung Wu            1912       1997 confim and refine the…
3 Katherine Johnson          1918       2020 calculations of orbit…
4 Vera Rubin                 1928       2016 existence of dark mat…
5 Gladys West                1930         NA mathematical modeling…
6 Flossie Wong-Staal         1947         NA first scientist to cl…
7 Jennifer Doudna            1964         NA one of the primary de…
```

Summary of Mutating Joins

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

Multiple matches? Return all combinations.

We want to get the dates and works of all the scientists. Which join function should we use?

```
# A tibble: 10 × 5
   name               profession  birth_year death_year known_for
   <chr>              <chr>            <int>      <int> <chr>    
 1 Ada Lovelace       Mathematic…         NA         NA first co…
 2 Marie Curie        Physicist …         NA         NA theory o…
 3 Janaki Ammal       Botanist          1897       1984 hybrid s…
 4 Chien-Shiung Wu    Physicist         1912       1997 confim a…
 5 Katherine Johnson  Mathematic…       1918       2020 calculat…
 6 Rosalind Franklin  Chemist           1920       1958 <NA>     
 7 Vera Rubin         Astronomer        1928       2016 existenc…
 8 Gladys West        Mathematic…       1930         NA mathemat…
 9 Flossie Wong-Staal Virologist…       1947         NA first sc…
10 Jennifer Doudna    Biochemist        1964         NA one of t…
```

```r
names(professions)
```

```
[1] "name"       "profession"
```

```r
names(dates)
```

```
[1] "name"       "birth_year" '
```

```r
names(notability)
```

```
[1] "name"      "known_for"
```

```r
nrow(professions)
```

```
[1] 10
```

```r
nrow(dates)
```

```
[1] 8
```

```r
nrow(notability)
```

```
[1] 9
```

```
professions %>%
  left_join(dates) %>%
  left_join(notability)
```

```
# A tibble: 10 × 5
   name               profession  birth_year death_year known_for
   <chr>              <chr>            <int>      <int> <chr>    
 1 Ada Lovelace       Mathematic…         NA         NA first co…
 2 Marie Curie        Physicist …         NA         NA theory o…
 3 Janaki Ammal       Botanist          1897       1984 hybrid s…
 4 Chien-Shiung Wu    Physicist         1912       1997 confim a…
 5 Katherine Johnson  Mathematic…       1918       2020 calculat…
 6 Rosalind Franklin  Chemist           1920       1958 <NA>     
 7 Vera Rubin         Astronomer        1928       2016 existenc…
 8 Gladys West        Mathematic…       1930         NA mathemat…
 9 Flossie Wong-Staal Virologist…       1947         NA first sc…
10 Jennifer Doudna    Biochemist        1964         NA one of t…
```

Case study: Grocery sales

Grocery sales

- Have:
  - *Purchases*: One row per customer per item, listing purchases they made
  - *Prices*: One row per item in the store, listing their prices
- **Want**: Total revenue

```
purchases
```

customer_id	item
c1	bread
c1	milk
c1	banana
c2	milk
c2	toilet paper

```
prices
```

item	price
avocado	0.50
banana	0.15
bread	1.00
milk	0.80
toilet paper	3.00

Grocery sales

```
purchases %>%
  left_join(prices)
```

```
# A tibble: 5 × 3
  customer_id item         price
  <chr>       <chr>        <dbl>
1 c1          bread         1
2 c1          milk          0.8
3 c1          banana        0.15
4 c2          milk          0.8
5 c2          toilet paper  3
```

Grocery sales

Total revenue

Revenue per customer

```
purchases %>%
  left_join(prices)
```

```
# A tibble: 5 × 3
  customer_id item         pric
  <chr>       <chr>        <dbl
1 c1          bread         1
2 c1          milk          0.8
3 c1          banana        0.1
4 c2          milk          0.8
5 c2          toilet paper  3
```

```
purchases %>%
  left_join(prices) %>%
  summarize(total_revenue = su
```

```
# A tibble: 1 × 1
  total_revenue
          <dbl>
1          5.75
```

Extension: Multiple matching rows

professions_multi   dates   notability_multi

```
professions_multi
```

```
# A tibble: 12 × 2
  name               profession  
  <chr>              <chr>       
1 Ada Lovelace       Mathematician
2 Marie Curie        Physicist   
3 Marie Curie        Chemist     
4 Janaki Ammal       Botanist    
5 Chien-Shiung Wu    Physicist   
6 Katherine Johnson Mathematician
# … with 6 more rows
```

Other types of joins

These are less common:

filtering joins

- `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

- `nest_join()`: get bundles of all matching rows from `y` (most flexible)

Specifying keys

- Keys must match *exactly*
- Can join on multiple columns (first name **and** last name)
- Default join: columns with same names
- Specify what columns to use: `left_join(x, y, by = c("first_name", "last_name"))`

General notes

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)

Speaking of better names

> `select` vs `filter`?

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