



# Joining Geospatial Datasets



# Joining Geospatial Datasets



# Bind or join?

There are two different ways to combine datasets using the `{dplyr}` package.

**Binding** datasets combines tables by the **position** of rows.

The diagram illustrates the concept of binding datasets. It shows two separate tables, labeled X and y, positioned above a horizontal line. A large plus sign (+) is placed between them, indicating the operation of binding. Above the plus sign, the letter X is centered above the first table, and the letter y is centered below the second table. The tables themselves are as follows:

	A	B	C
a	t	1	
b	u	2	
c	v	3	

  

	A	B	C
C	v	3	
d	w	4	

**Joining** datasets combines tables by the **values** in rows.

The diagram illustrates the concept of joining datasets. It shows two tables, x and y, positioned above a plus sign (+). To the right of the plus sign is an equals sign (=), indicating the result of the joining operation. The tables are as follows:

	A	B	C
a	t	1	
b	u	2	
c	v	3	

  

	A	B	D
a	t	3	
b	u	2	
d	w	1	

The resulting joined table is:

	A	B	C	A	B	D
a	t	1		a	t	3
b	u	2		b	u	2
c	v	3		d	w	1



# "Mutating joins"

The `{dplyr}` documentation includes one page called "mutate-joins" that covers four **different kinds of join**:

- `left_join()`
- `right_join()`
- `inner_join()`
- `full_join()`

`{tidyexplain}` contains GIF that explain each of these graphically:



# More about mutating joins...

`left_join()` is the most common (and usually correct) join.

Claus Wilke @ClausWilke · Mar 4  
Exactly.

Andrew Whitby @EconAndrew · Mar 4  
Replying to @StatStas and @beeonapossy  
The right join is left join.

1 1 14

Claus Wilke @ClausWilke · Mar 4  
When teaching the other day, I found myself saying: Always assume the left join is the right join, and just deal with it if you realize it's not.

11

Learn more about the mechanics of joining on R for the Rest of Us.





# Joining sf objects

Always place the {sf} dataset in the first argument of the join functions:

```
uk_sf <- read_sf("data/shapefiles")
uk_population <- read_csv("data/uk_population.csv")
```

## Example Join

```
left_join(uk_sf, uk_population)
```



```
left_join(uk_population, uk_sf)
```



```
right_join(uk_sf, uk_population)
```



```
right_join(uk_population, uk_sf)
```



## Keeps geometry?



# One last thing...

In the documentation the two arguments of all the joins and binding functions are named `x` and `y`:

- `left_join(x, y)`
- `right_join(x, y)`
- `full_join(x, y)`
- `inner_join(x, y)`
- `bind_cols(x, y)`
- `bind_rows(x, y)`

You might prefer to think of them as the **left-hand side** and **right-hand side** of the joins.



# My Turn

I'm going to show how to use `left_join()` to combine datasets from the `{gapminder}` and `{rnaturalearthdata}` datasets.

I'll then show you how to bind multiple `{sf}` objects together.



# (RStudio Coding Slide)



# Your Turn

Use `mapview()` to visualise the results of the Brexit Referendum by combining shapefiles with the a .csv file containing the results.

- Combine together the shapefiles for England, Scotland and Wales with `bind_rows()`
- Import the referendum results .csv file
- Use `left_join()` to join the UK shapefiles with the referendum results
- Use `mapview(zcol = "results")` to visualise the shapefiles after joining