Mutating Joins

Muhammad Yaseen

Contents

- Introduction
- Inner Join
- Outer Joins

Introduction

Introduction

- *Mutating Joins* **add columns** from y to x, matching observations based on the *keys*.
- There are *Four Mutating Joins*:
  - *inner join*
    - `inner_join()`
  - *three outer joins*
    - `left_join()`
    - `right_join()`
    - `full_join()`

Introduction

- An `inner_join()` only keeps observations from `x` that have a matching key in `y`.
- The most important property of an inner join is that unmatched rows in either input are not included in the result.
- This means that generally inner joins are not appropriate in most analyses, because it is too easy to lose observations.

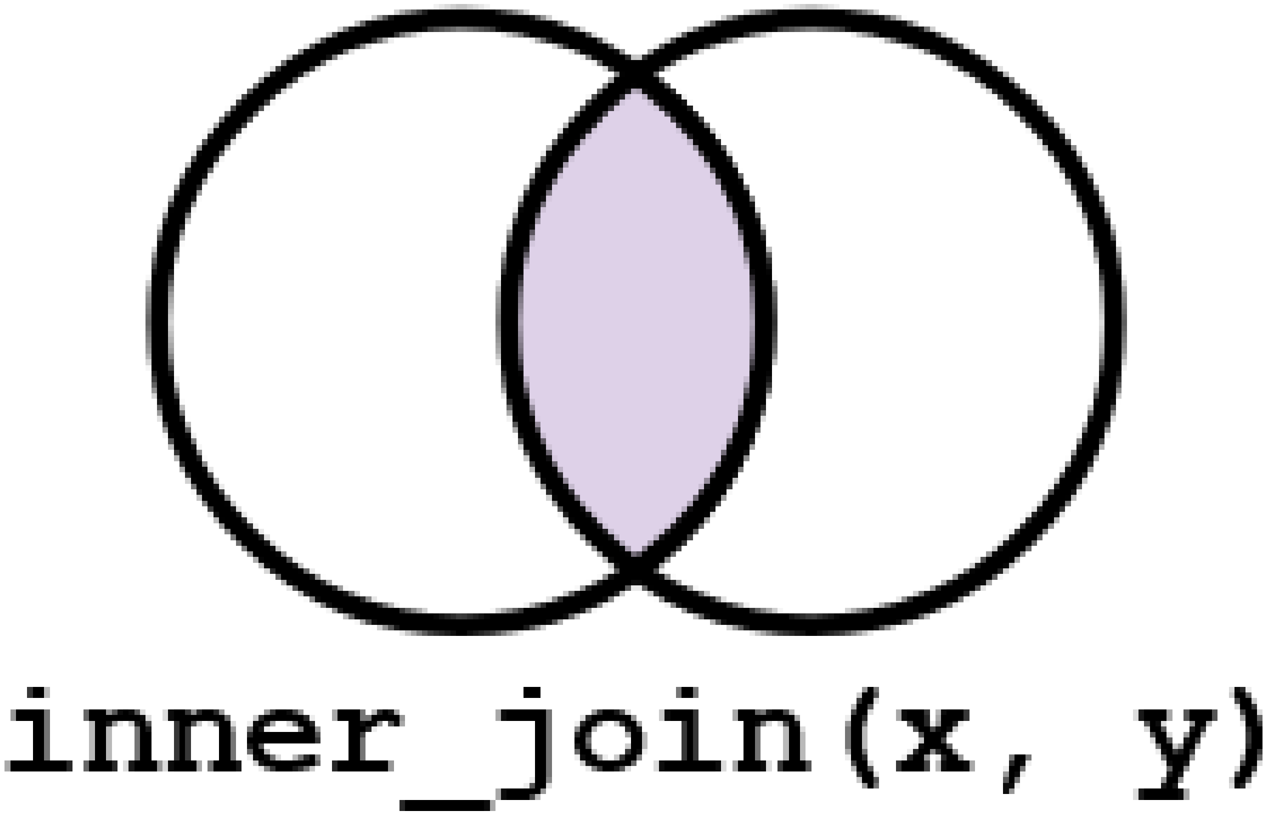
Introduction

- The three outer joins keep observations that appear in at least one of the data frames:
  - A `left_join()` keeps all observations in `x`.
  - A `right_join()` keeps all observations in `y`.
  - A `full_join()` keeps all observations in `x` and `y`.

Inner Join

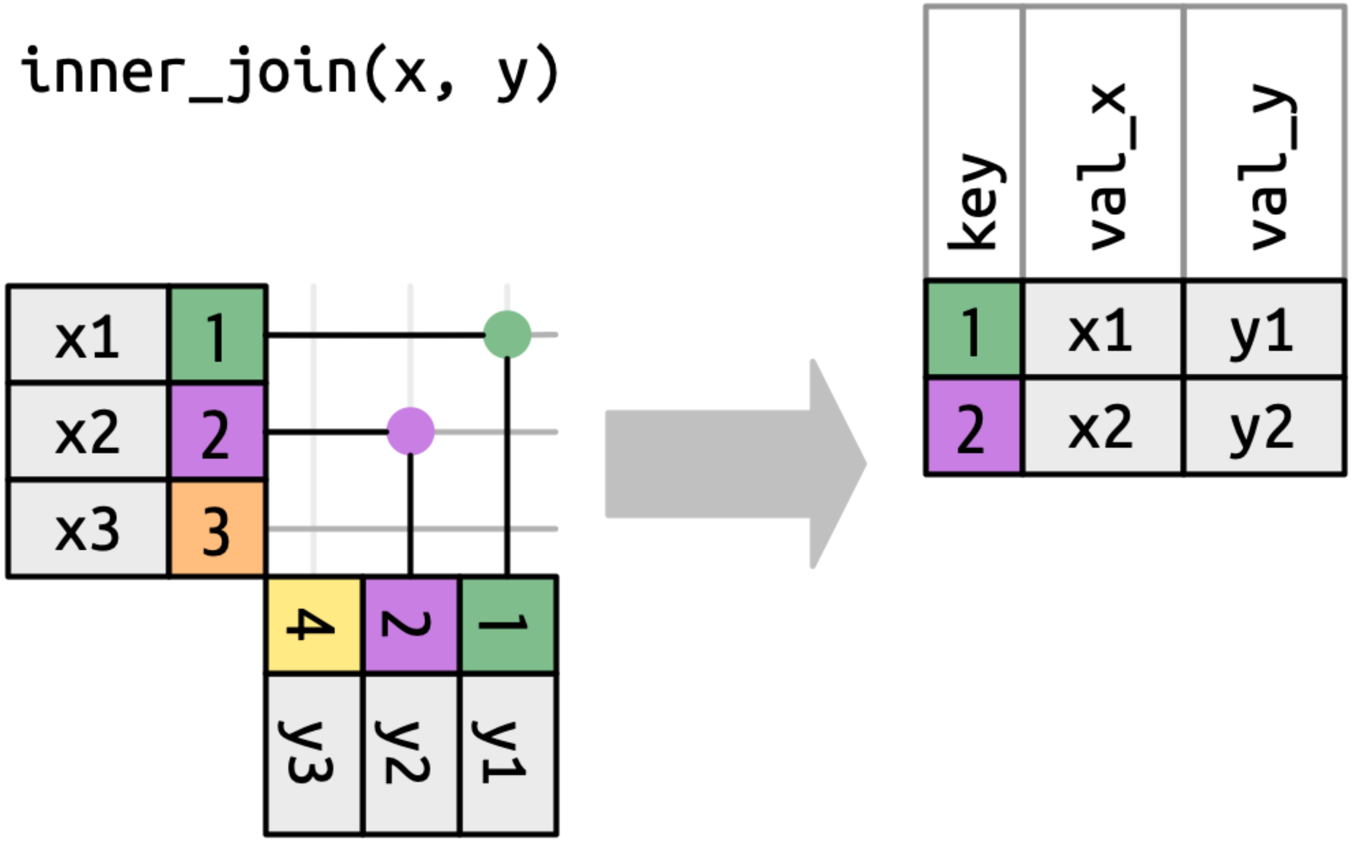
Inner Join

- An `inner_join()` only keeps observations from `x` that have a matching key in `y`.



Inner Join

- An `inner_join()` only keeps observations from `x` that have a matching key in `y`.



Inner Join

Output   Code

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

```
# A tibble: 3 × 2
    key val_y
  <dbl> <chr>
1     1 y1
2     2 y2
3     4 y3
```

Inner Join

```
inner_join(
           x,
           y,
           by = NULL,
           copy = FALSE,
           suffix = c(".x", ".y"),
           ...,
           keep = NULL)
```

Inner Join

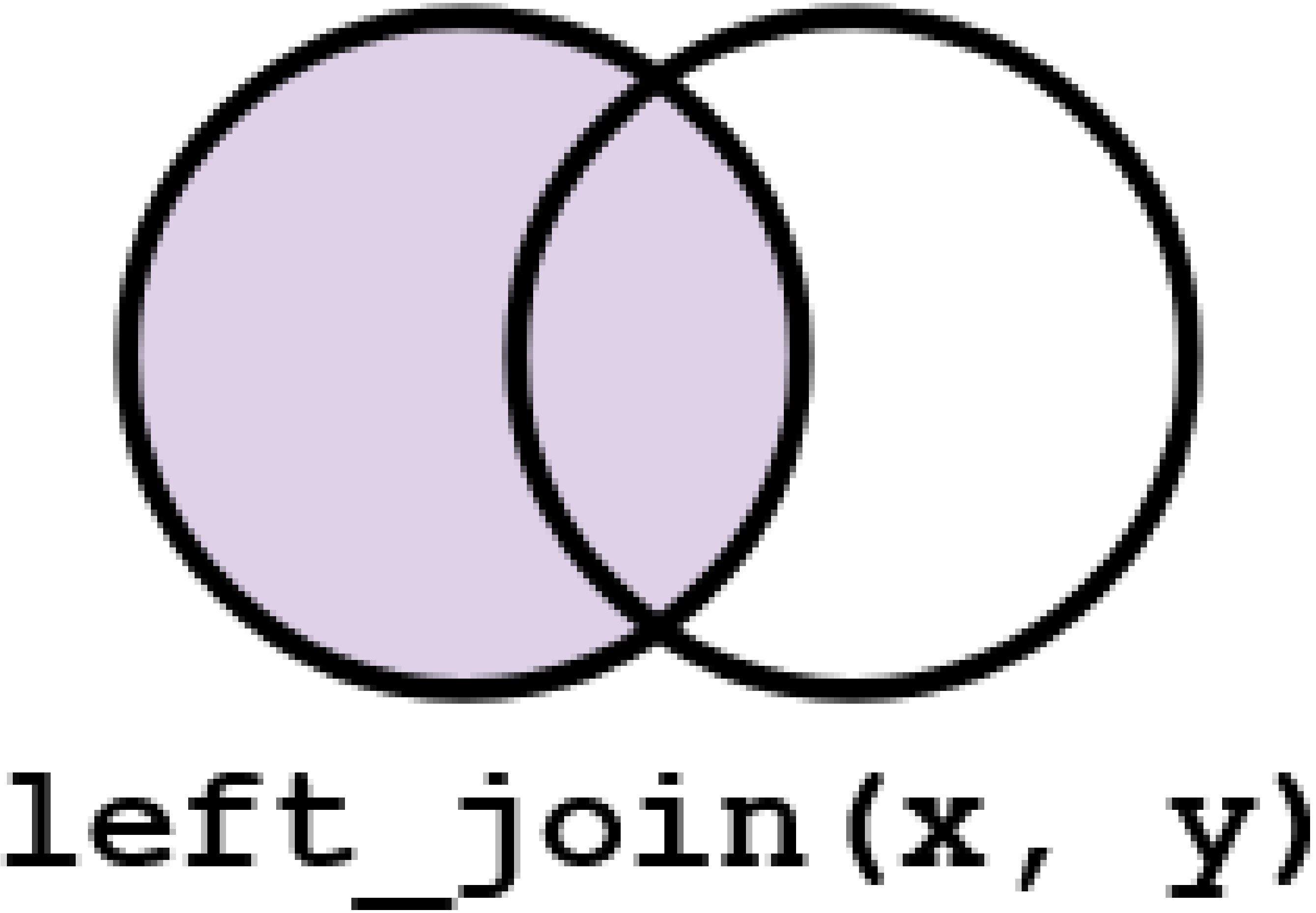
Output Code

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

Outer Joins

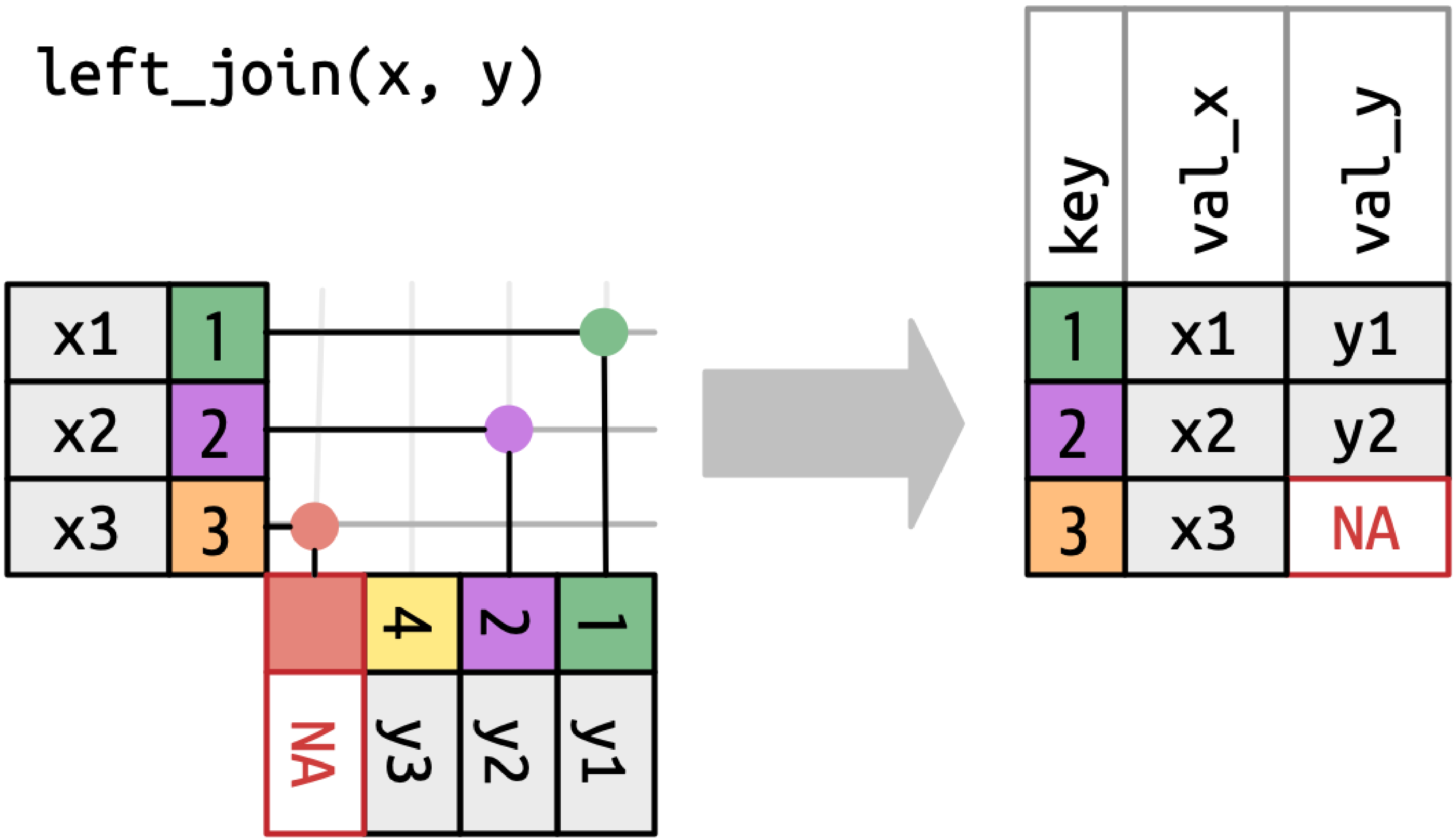
Left Join

- An `left_join()` keeps all observations in `x`.



Left Join

- An `left_join()` keeps all observations in `x`.



Left Join

```
left_join(x,
          y,
          by = NULL,
          copy = FALSE,
          suffix = c(".x", ".y"),
          ...,
          keep = NULL)
```

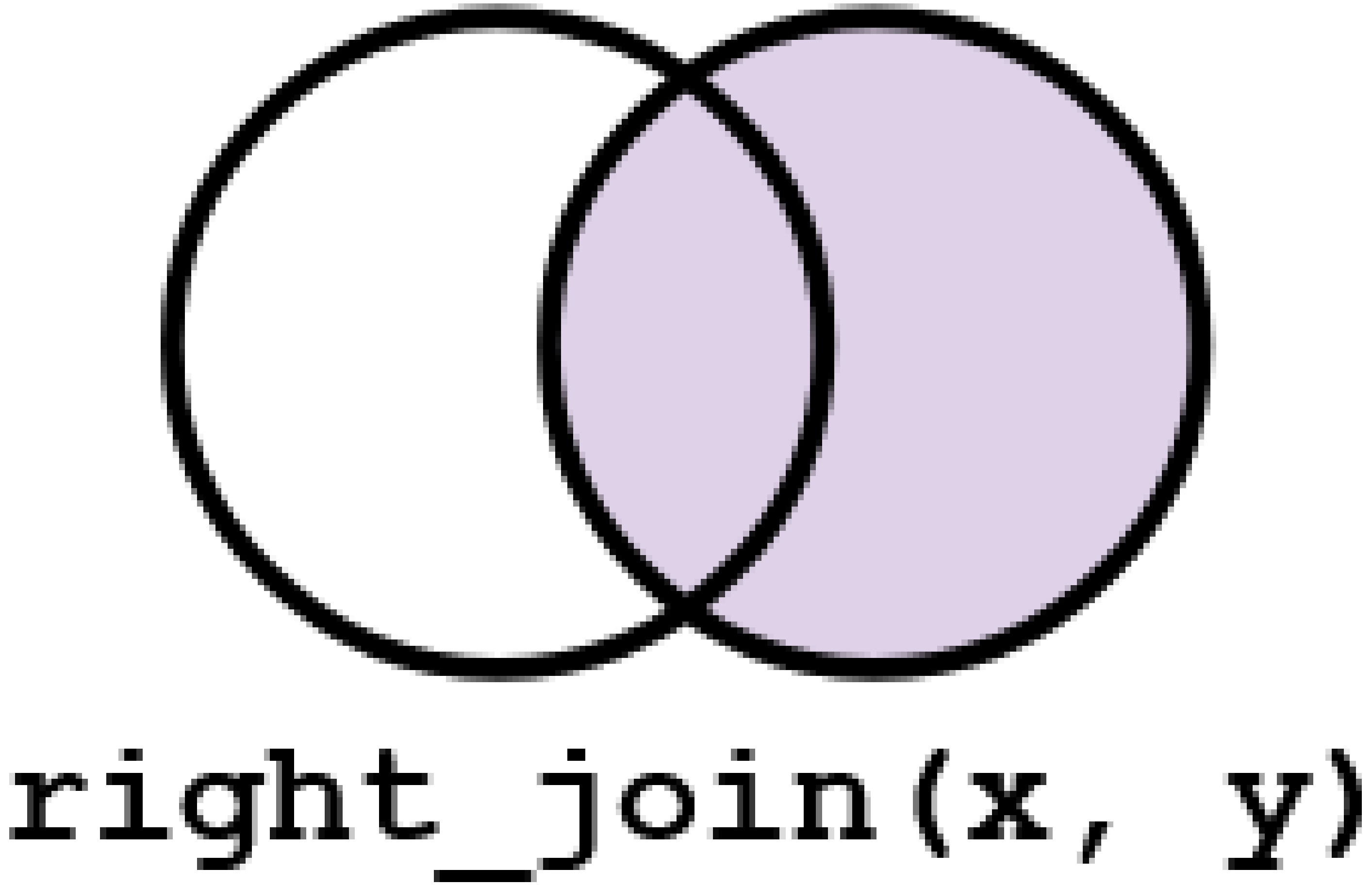
Left Join

Output Code

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

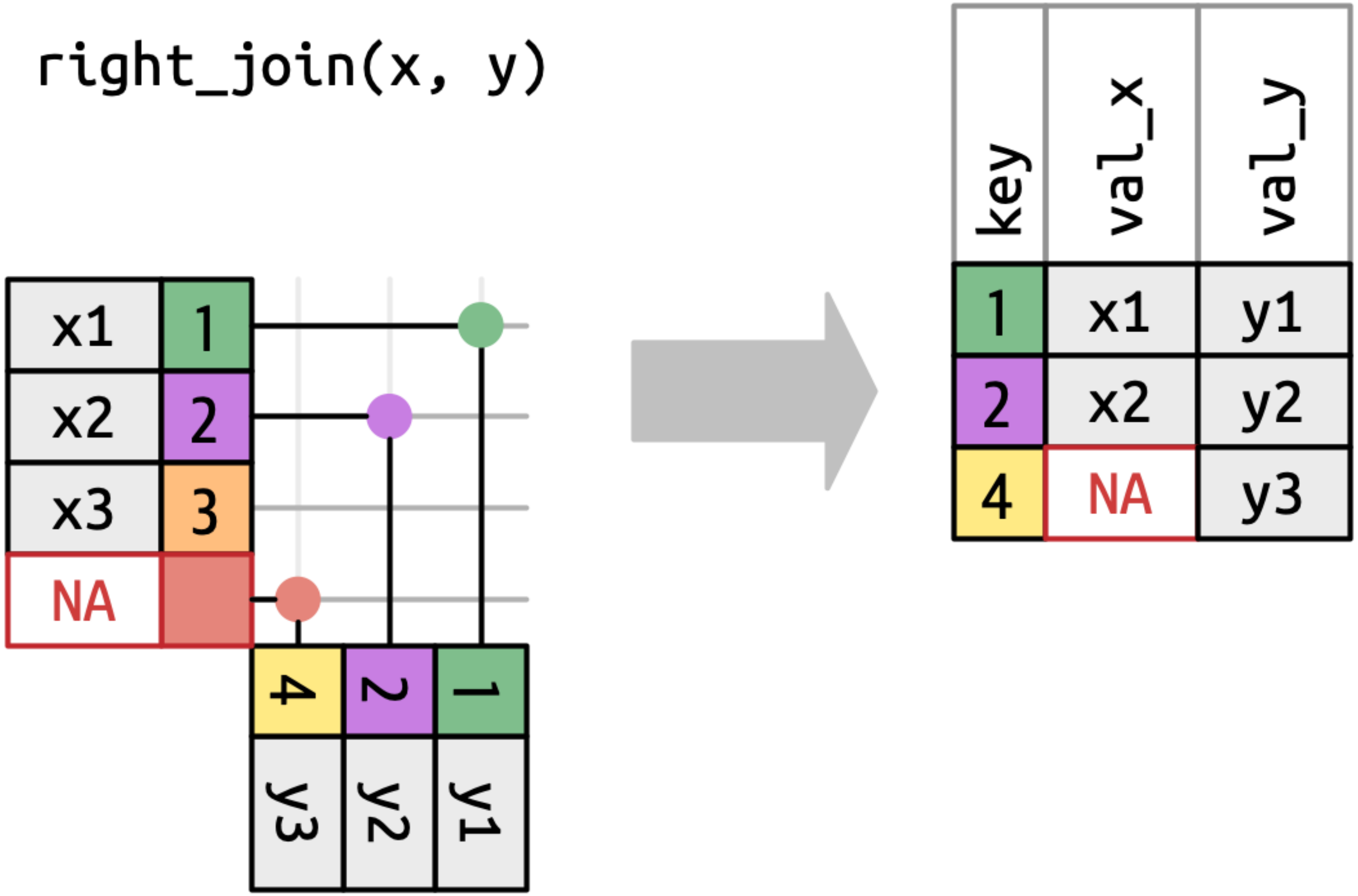
Right Join

- An `right_join()` keeps all observations in `y`.



Right Join

- An `right_join()` keeps all observations in `y`.



right_join(x, y)

key	val_x	val_y
1	x1	y1
2	x2	y2
4	NA	y3

Right Join

```
right_join(
           x,
           y,
           by = NULL,
           copy = FALSE,
           suffix = c(".x", ".y"),
           ...,
           keep = NULL)
```

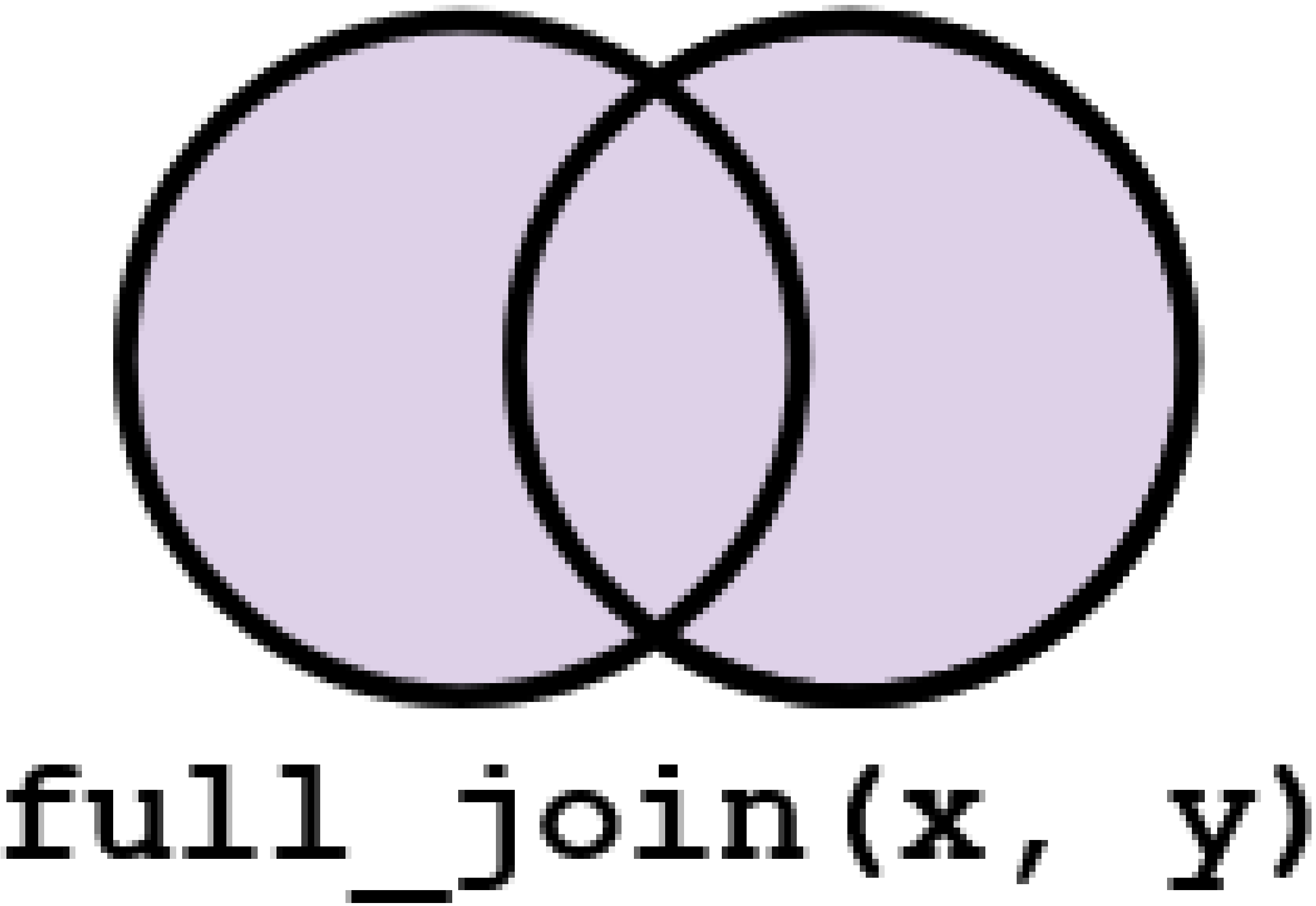
Right Join

Output Code

```
# A tibble: 3 × 3
    key val_x val_y
  <dbl> <chr> <chr>
1     1 x1    y1
2     2 x2    y2
3     4 <NA>  y3
```

Full Join

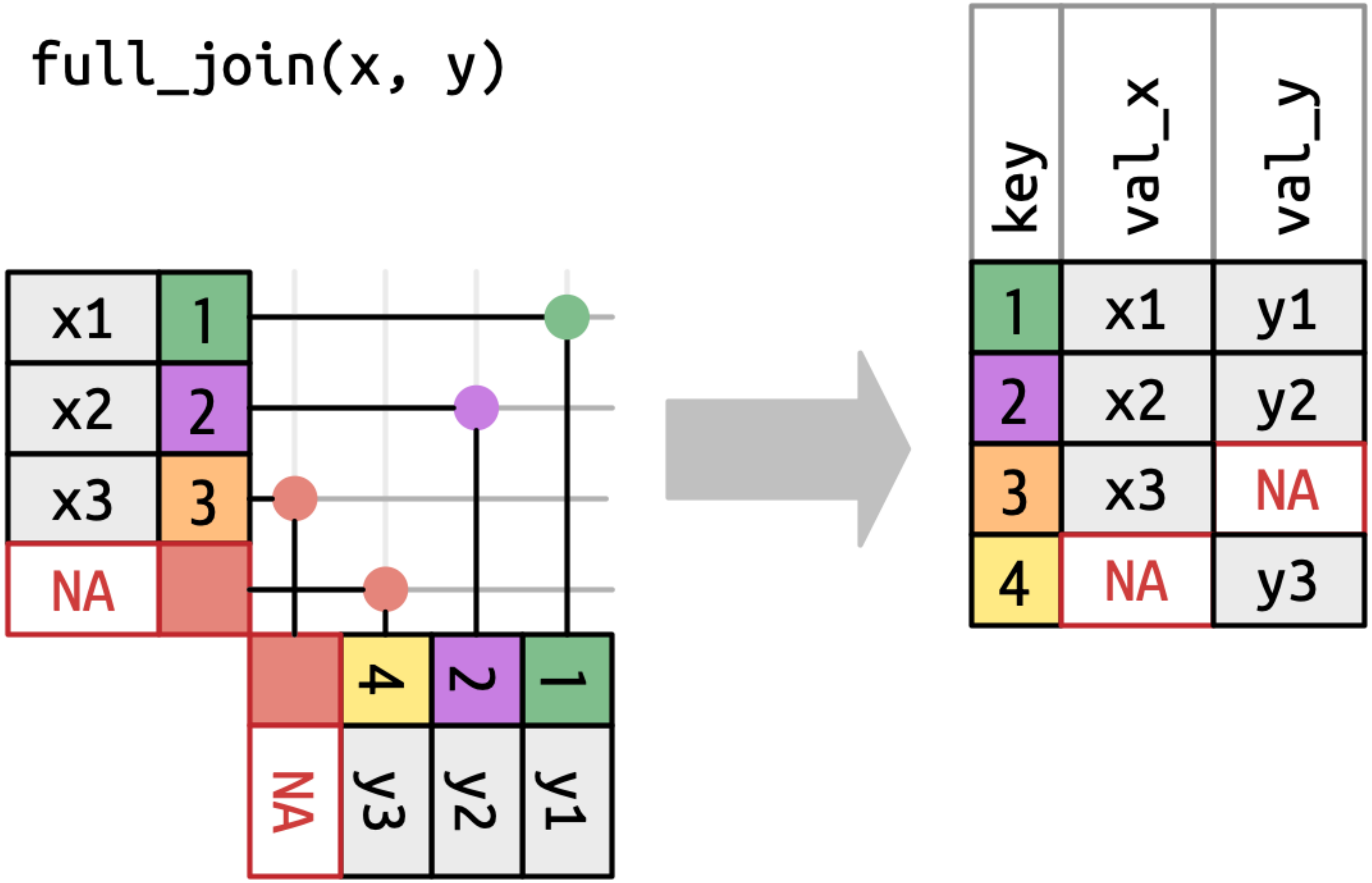
- An `full_join()` keeps all observations in `x` and `y`.



Full Join

- An `full_join()` keeps all observations in `x` and `y`.

`full_join(x, y)`



key	val_x	val_y
1	x1	y1
2	x2	y2
3	x3	NA
4	NA	y3

Full Join

```
full_join(x,
          y,
          by = NULL,
          copy = FALSE,
          suffix = c(".x", ".y"),
          ...,
          keep = NULL)
```

Full Join

Output Code

```
# A tibble: 4 × 3
    key val_x val_y
  <dbl> <chr> <chr>
1     1 x1    y1
2     2 x2    y2
3     3 x3    <NA>
4     4 <NA>  y3
```