Foundations of Data Science for Biologists

Data wrangling: merges and joins

BIO 724D

2024-SEP-25

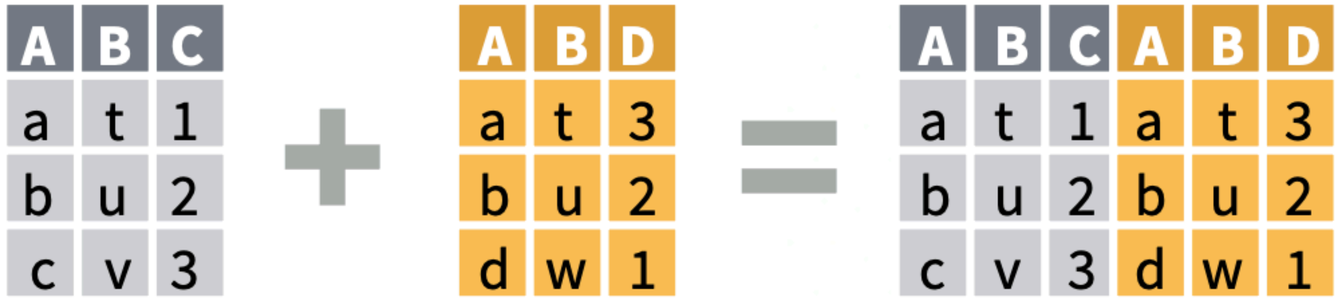
Instructors: Greg Wray, Paul Magwene

What are merge and join operations?

Both are methods for combining information from two data frames

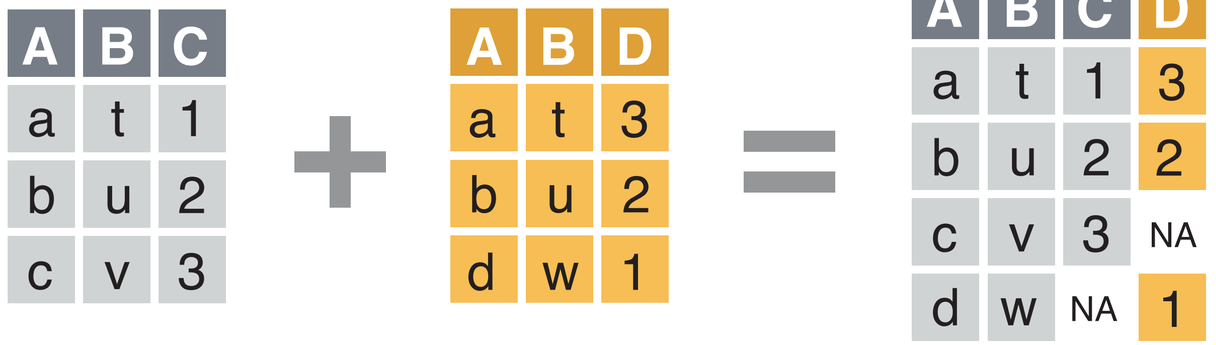
They involve different processes and have distinct uses

Merge



Based on order

Join

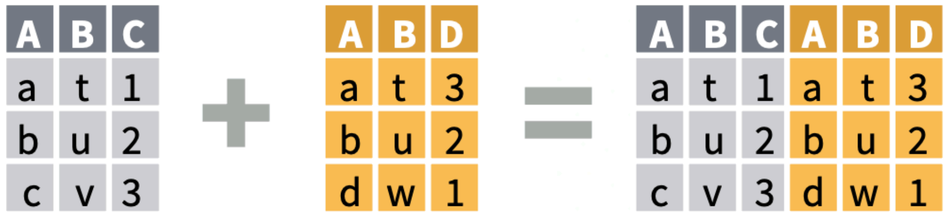


Based on values

Data frame merges

Merge operations

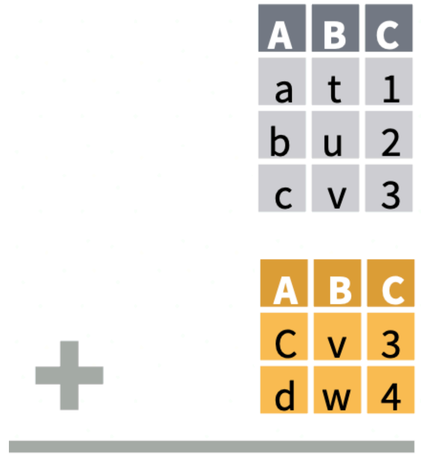
Merge columns

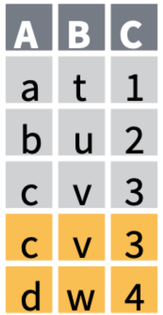


Add more observations (wide-form)

Add more attributes (long-form)

Merge rows





Add more observations (long-form)

Considerations when merging **columns**

Check that the two data frames are compatible before you try to merge

- Check for equivalent number of rows using the function `nrow()`
- If adding additional observations, make sure rows align correctly
- If adding additional attributes, makes sure values match the correct rows

During the merge, column names are checked and repaired by default

- Duplicates are re-named and names are added if not present

All data are retained without modification or filtering

- Columns with the same name and columns with identical values are all kept

Warning: `dplyr` merge functions are quite forgiving

- It is up to you to make certain that the merge makes sense!

Considerations when merging **rows**

Check that the two data frames are compatible before you try to merge

- The two data frames should have the same columns (meaning, not just name)
- Columns are matched by name, not by position
- Any missing column will be filled with `NA` values
- Check for the expected number of columns after the merge using `length()`

Decisions about how to merge:

- Which set logic to use (see next slide)
- Whether you want to retain information about table of origin (see documentation)

Warning: `dplyr` merge functions are quite forgiving

- It is up to you to ensure that the merge makes sense!

Data frame joins

What is a join? How does it differ from a merge?

Join: combine data in two tables based on values in 1 (or more) columns

- Row order doesn't matter!
- Values in the join column(s) dictate how data are combined
- Best way to understand is with diagrams on following slides

Process of combining data differs:

- **Merge**: attach rows or columns in order, ignoring values
- **Join**: attach data based on values in the join column(s), ignoring order

Outcome thus differs:

- **Merge**: all data are preserved; row and column order is preserved
- **Join**: some data are removed; row and column order are typically rearranged

Specifying a join

Joins rely on matching values in one or more **columns** present in both data frames

- Typically, just one column is matched (e.g., a unique ID)
- However, matching in multiple columns is possible (e.g., genus and species)

Decisions about how to join

- Set logic determines which rows and columns are retained and how they are attached
- Duplicate rows can be retained or discarded

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

+

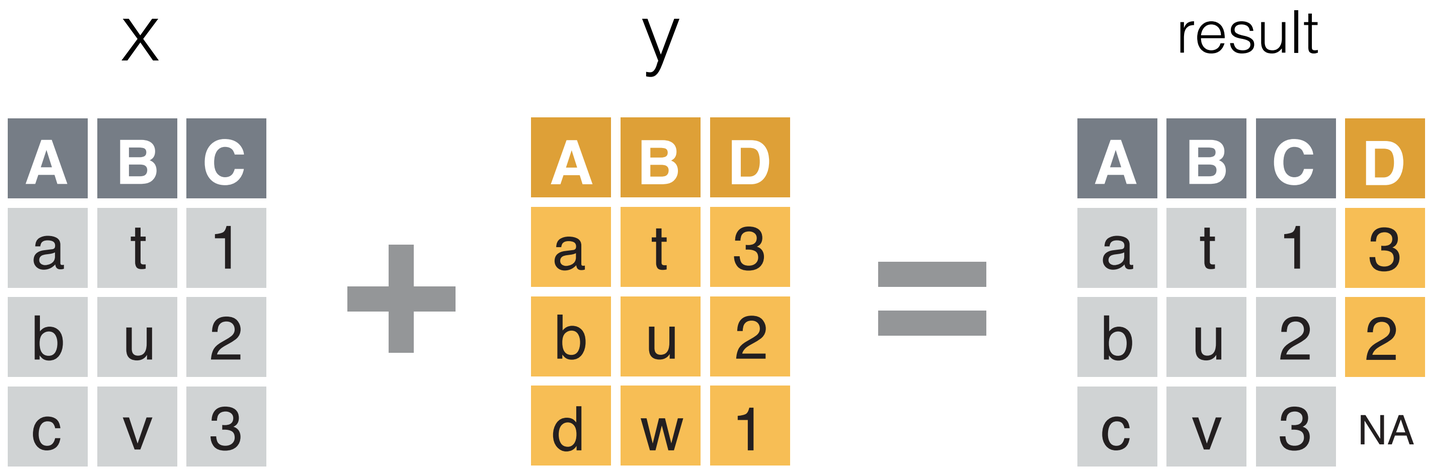
A	B	D
a	t	3
b	u	2
d	w	1

The following examples join on column **A** only

This is different from default dplyr behavior

Left join

Left joins (also known as left outer joins) are a very common type of join operation

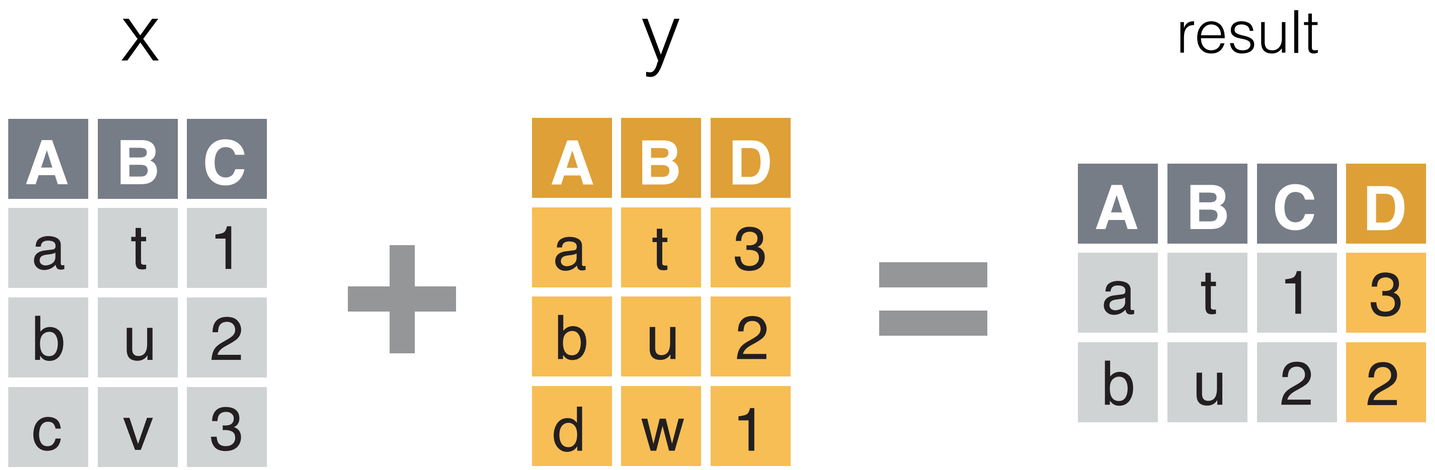


Left joins create a union of matches to the first (left) data frame: order matters!

- Keep all rows from x
- Discard rows in y that do not match x
- Attach columns from y when a match (NA if not)
- Discard columns from y that duplicate those in x

Inner join

Inner joins are another common type of join operation

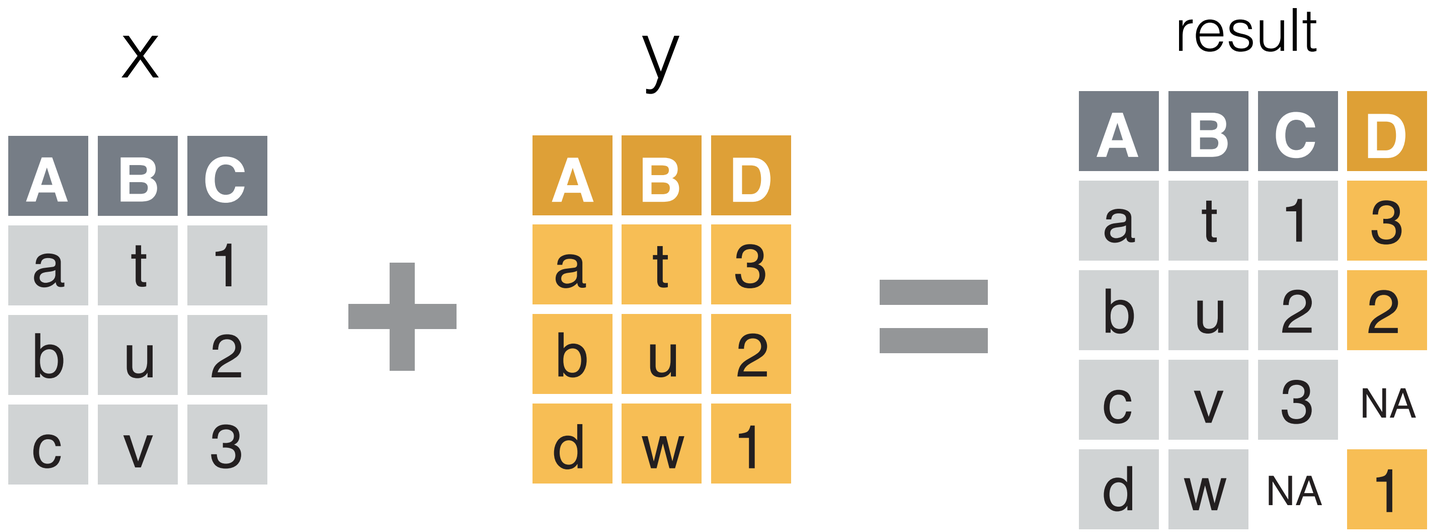


Inner joins create an intersection of matches between data frames: order does not matter

- Keep rows in x that match y
- Discard rows from x that do not match y
- Attach columns from y when a match
- Discard columns from y that duplicate those in x

Full join

Full joins are less common, but useful in specific situations



Full joins create a union of all rows and columns

- Keep all rows in x
- Add all rows from y that do not match x
- Attach columns from y when a match
- Insert NA where rows from y lack a column from x, and vice-versa

Comparing join operations

Source tables

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

+

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

Join methods

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

Left join

Union of matches to left (table x)

A	B	C	D
a	t	1	3
b	u	2	2

Inner join

Intersection of matches

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

Full join

Union of all rows and columns