R Lesson 5: Data Wrangling (part 2)

vanderbi.lt/r

Steve Baskauf

Download ICPSR data for later

- Instructions link on the lessons homepage (vanderbi.lt/r)
- Create ICPSR account
- Download 2 files for National Longitudinal Study of Adolescent to Adult Health, 1994-2008 (**`21600-0001-Data.tsv`** and **`21600-0022-Data.tsv`**)

Modifying tibbles (dplyr)

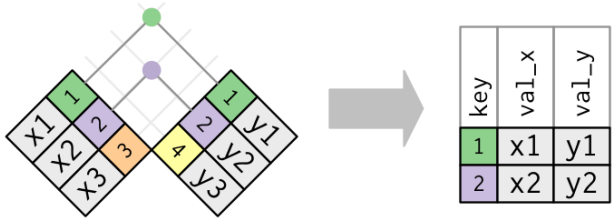
dplyr functions

- **`filter()`** subsets rows
- **`select()`** subsets columns
- **`mutate()`** calculates new columns or changes existing ones

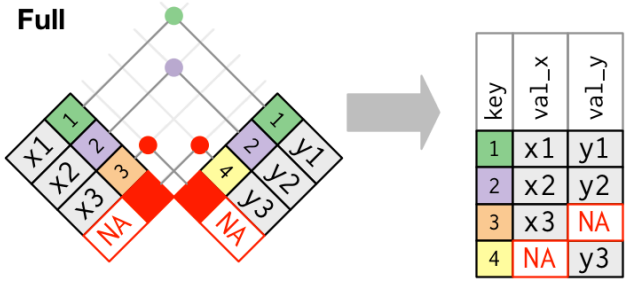
Examples with schools data

Joins

- Joins merge data from multiple tables (tibbles)
- **Keys** are the columns used to match table rows
- **Inner join** only outputs rows with matching keys



- **Full outer join** includes rows that don't match (with NA values inserted)



- Many other permutations
- See https://r4ds.had.co.nz/relational-data.html for explanation and examples (diagrams from there)

Join format

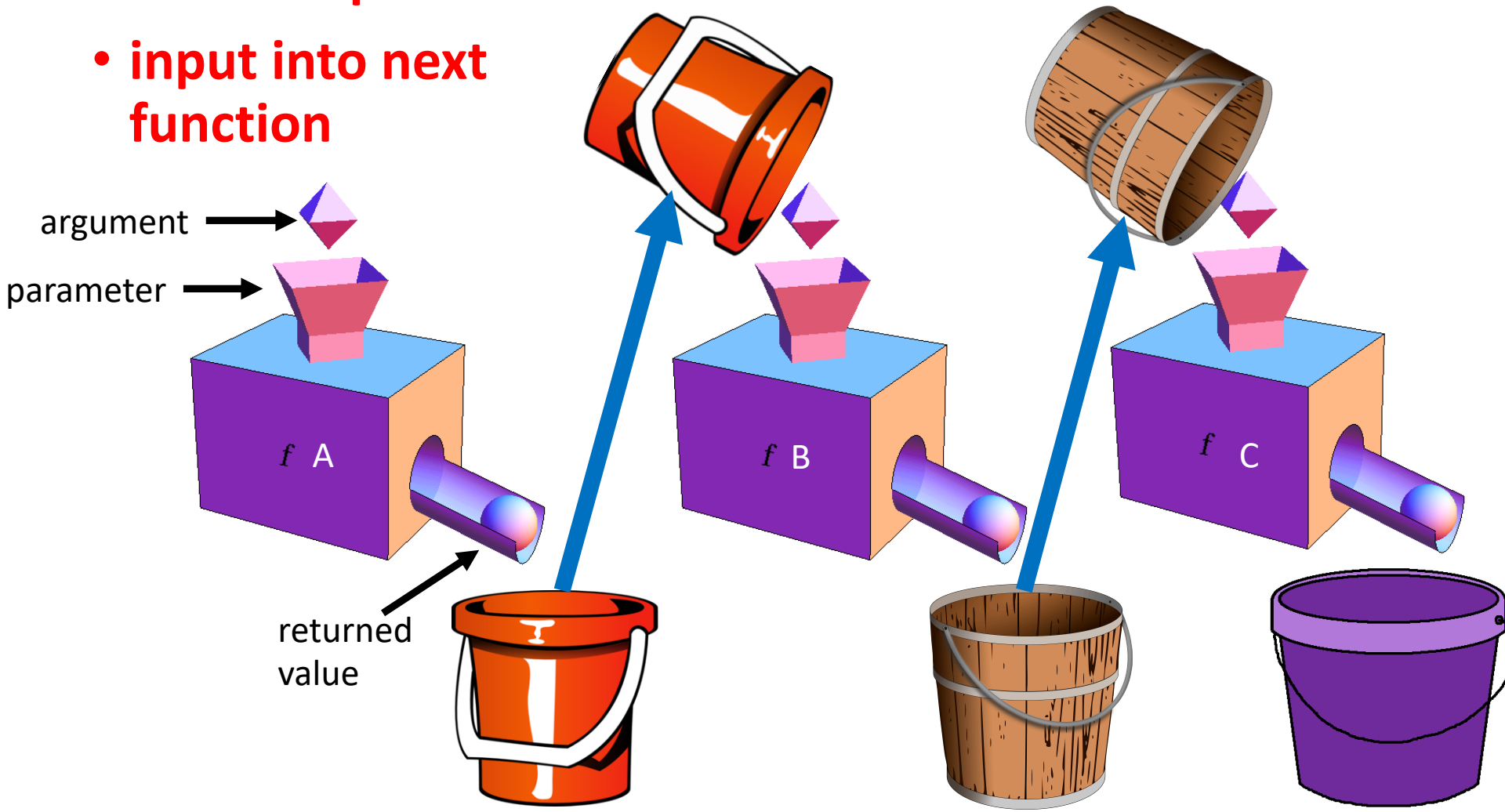
```
full_join(womens_data, poverty_data,
          by = c("country"="Country Name"),
          copy = FALSE,
          suffix = c(".wom", ".pov") )
```

- First two arguments are the two tibbles to join
- **`by`** value are columns to join by; use = if names differ
- **`suffix`** value is added to columns with duplicate names
- other join types: **`inner_join()`**, **`left_join()`**, …

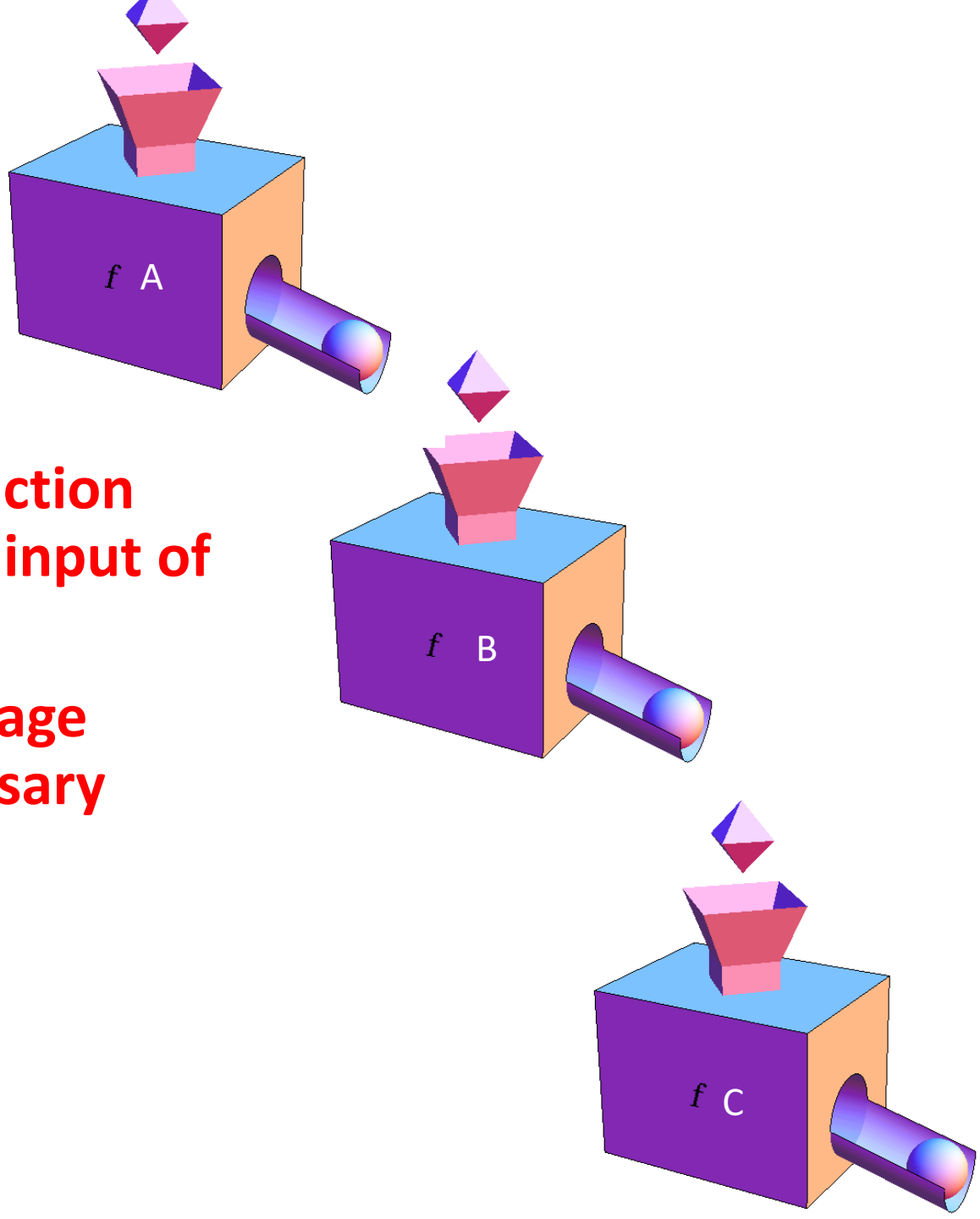
piplines (magrittr)

Classic function/variable interaction

- **store output**
- **input into next function**



Piping

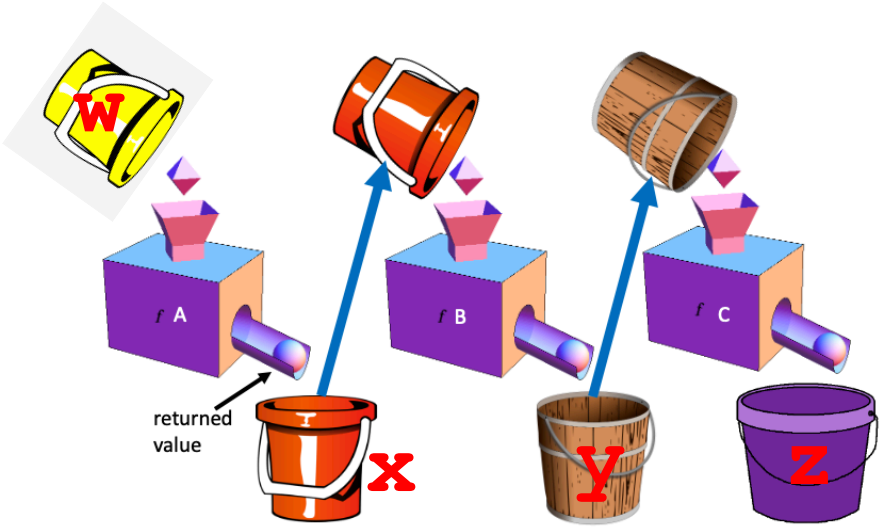


- **output of one function goes directly into input of next**
- **intermediate storage objects not necessary**

Examples

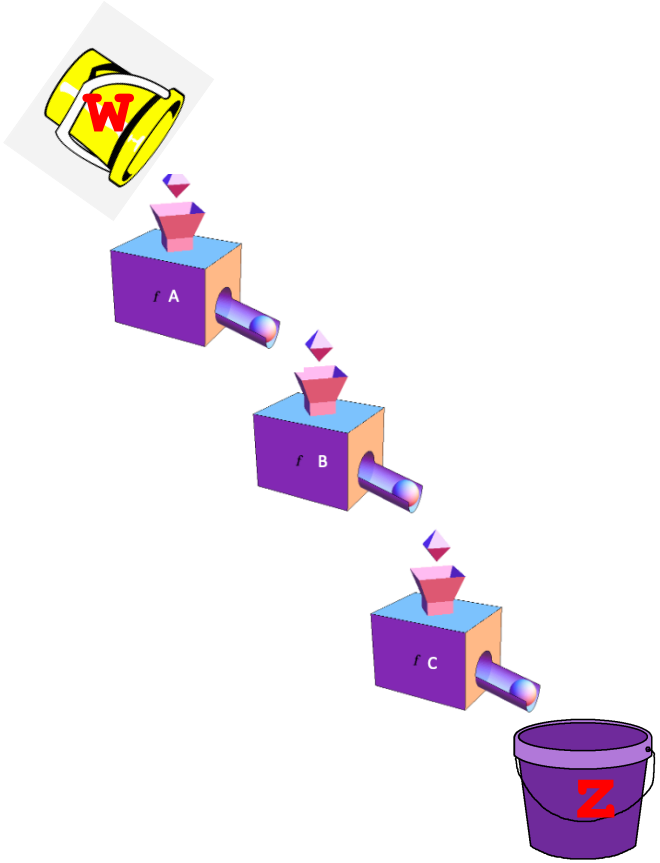
- Classic

```
x <- function_a(w, p)
y <- function_b(x, q)
z <- function_c(y, r)
```



- Piping

```
z <- w %>%
     function_a(p) %>%
     function_b(q) %>%
     function_c(r)
```



- Notice that no intermediate storage object needs to be input into the piped function

Examples with schools data