

# R Lesson 5: Data Wrangling (part 2)

[vanderbi.lt/r](http://vanderbi.lt/r)

Steve Baskauf



# Modifying tibbles (dplyr)



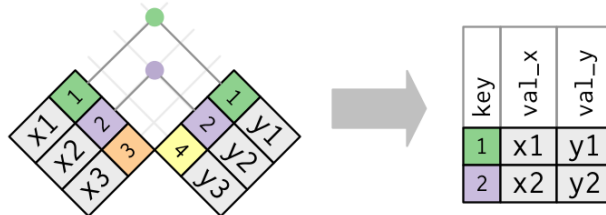
# dplyr functions

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

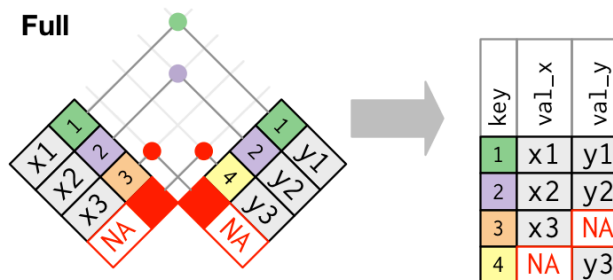
# Examples with schools data

# 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") )
```

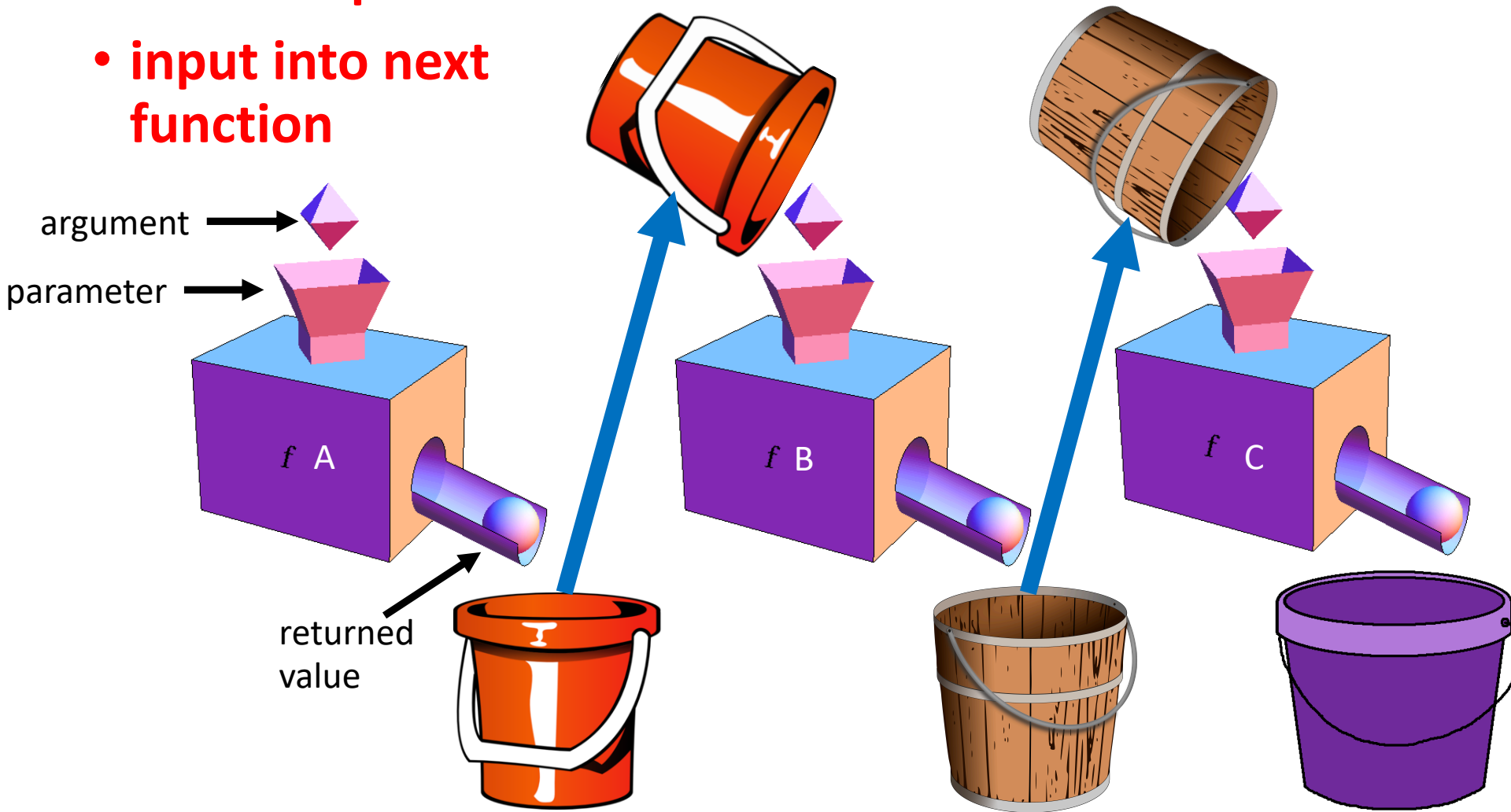
- First two arguments are the two tibbles to join
- **by** value are columns to join by; use = if names differ
- **suffix** value is added to columns with duplicate names
- other join types: **inner\_join()**, **left\_join()**, ...

pipelines (magrittr)



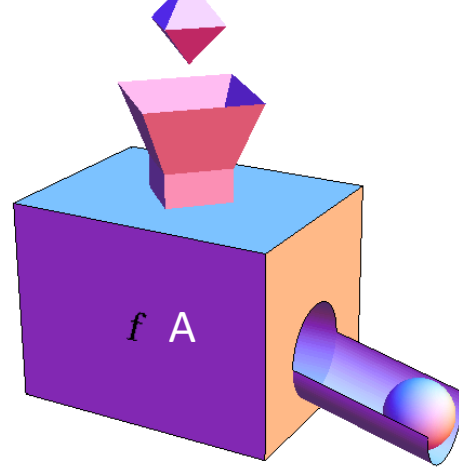
# Classic function/variable interaction

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

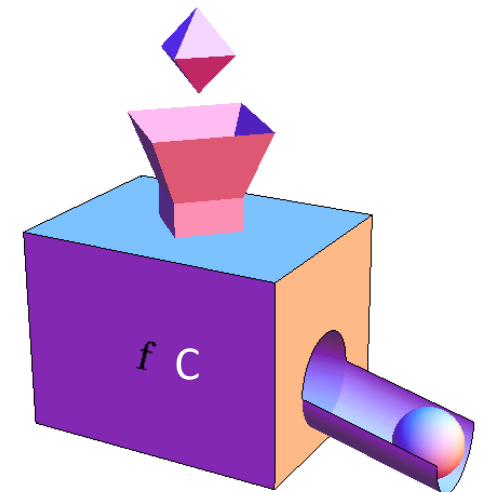
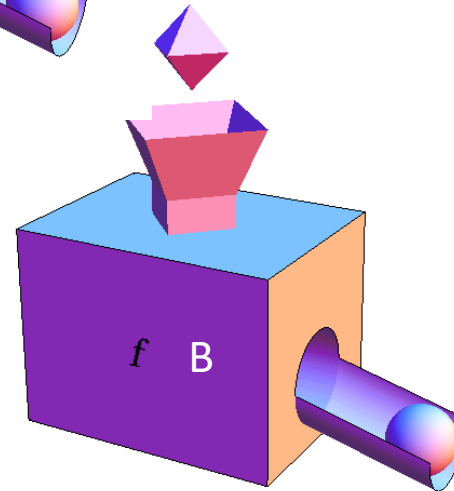




# Piping



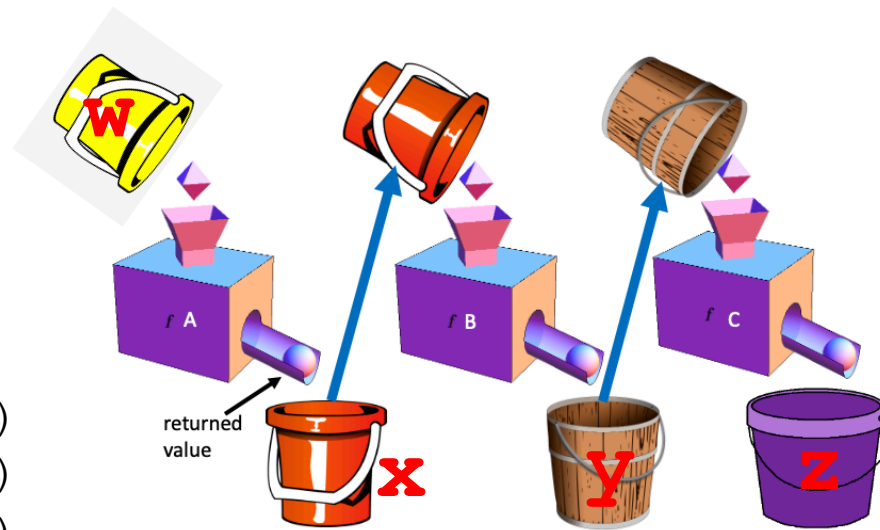
- **output of one function goes directly into input of next**
- **intermediate storage objects not necessary**



# Examples

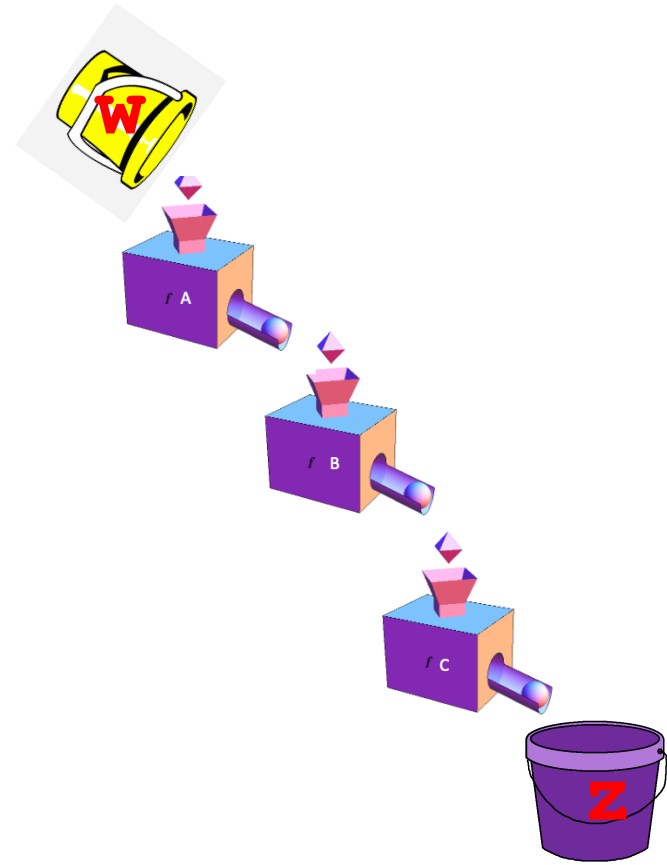
- Classic

```
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