

More data wrangling and piping

Presenter: Steve Baskauf
steve.baskauf@vanderbilt.edu



Jean & Alexander Heard
LIBRARIES

CodeGraf landing page

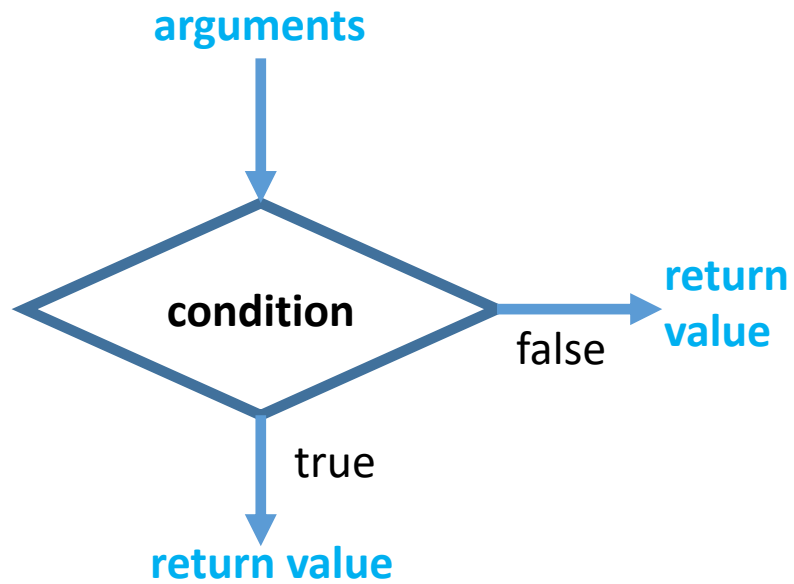
- vanderbi.it/codegraf

Changing tibbles (dplyr package)

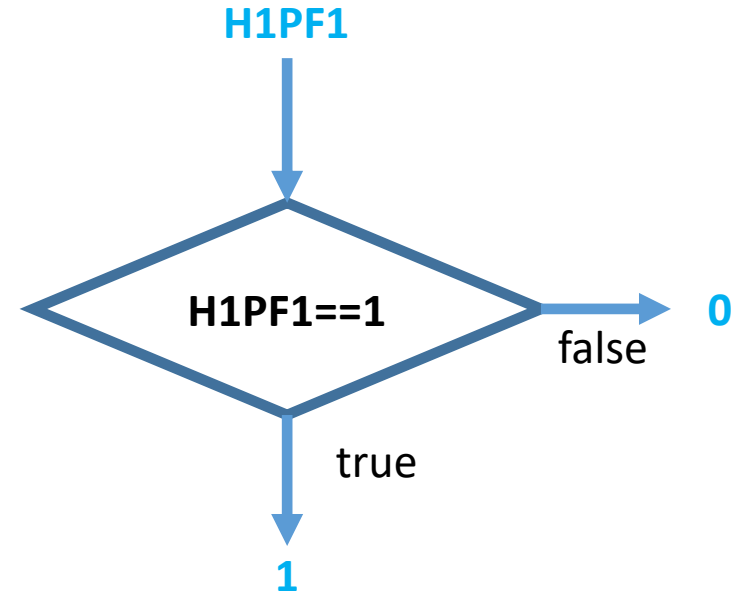
functions

- **filter()** subsets rows
- **select()** subsets columns
- **mutate()** creates new columns
- **transmute()** creates new tibbles by calculation
- ✓ **replace()** conditionally replaces values in a vector

`ifelse()` function



general pattern



specific example

```
ifelse(H1PF1==1, 1, 0)
```

boolean operators for conditions

! is **NOT**

& is **AND**

| is **OR**

- **Examples:**

`!H1PF5==1` H1PF5 isn't equal to 1

`!is.na(H1PF5)` H1PF5 doesn't have an NA value

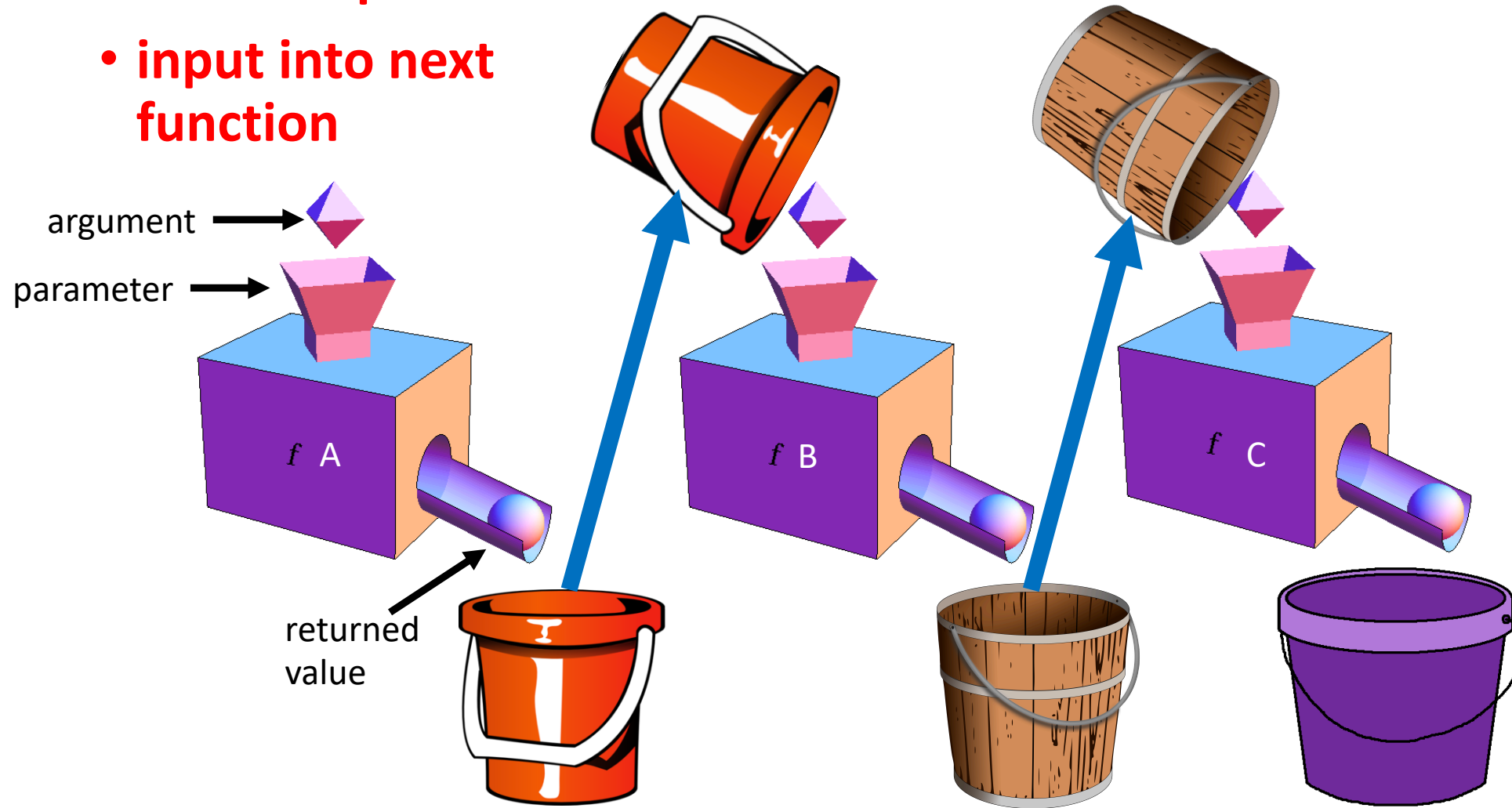
`H1PF3==1 & H1PF4==1` both H1PF3 and H1PF4 equal 1

`is.na(H1PF2) | is.na(H1PF3)` either H1PF2 or H1PF3 is NA

Pipelines (magrittr package)

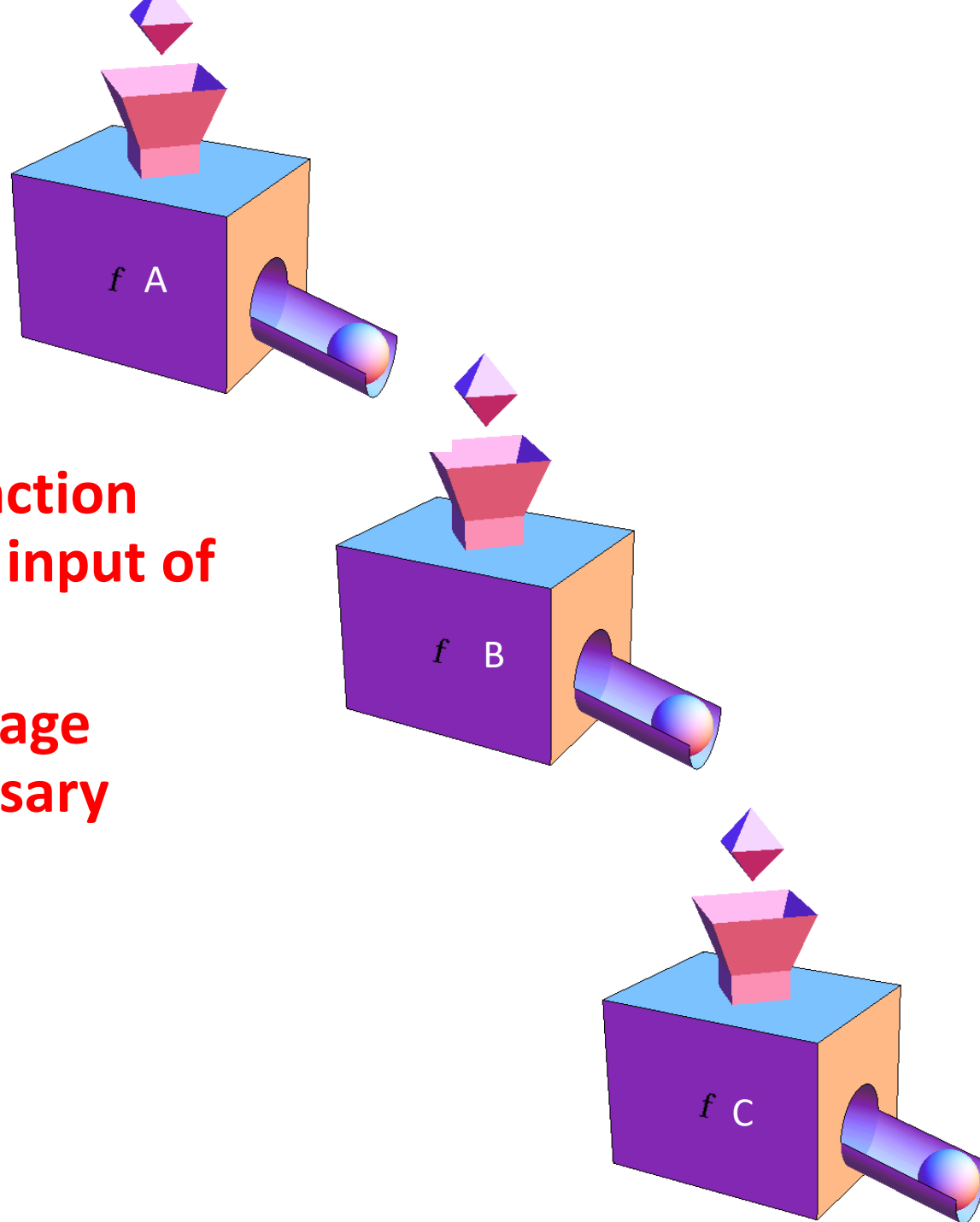
Classic function/variable interaction

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



Piping

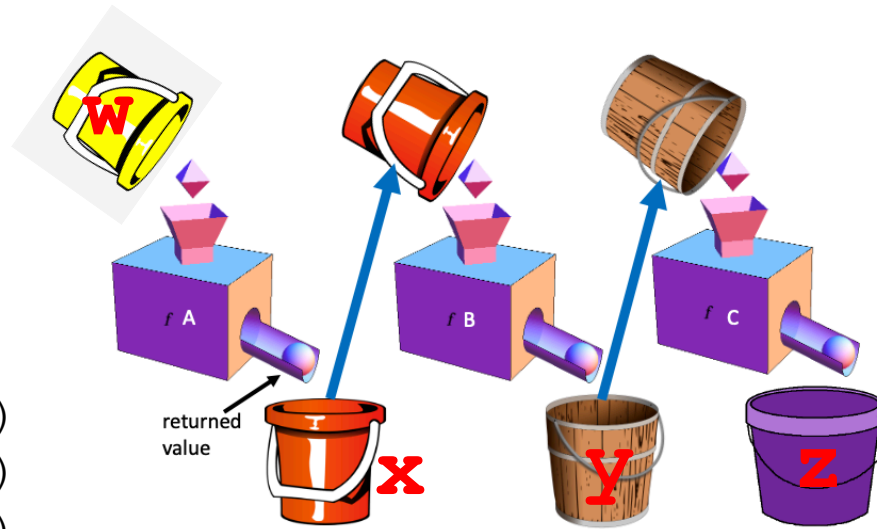
- **output of one function goes directly into input of next**
- **intermediate storage objects not necessary**



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

