

# R Lesson 4: Data Wrangling

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

Steve Baskauf



# Recommended for this lesson:

- URLs in R script for this week
- R For Data Science (free online); chapter links in script
- Data Carpentries lesson "Manipulating, analyzing and exporting data with tidyverse" (also free online)

# Options for recording data



# Cockroach electroretinogram experiment



- See <https://youtu.be/aAdnZsggZZw>
- Difference in ability to detect colors of light

# Experimental design

- two factors:
  - **color** (red, green, or blue)
  - **block** (24 individual roach measurements labeled a through x)
- one measured value (**response** in volts)
- How to record in notebook (or Excel)?

# Logical method

- columns for color
- rows for roach measured

	A	B	C	D	
1	block	blue	green	red	
2	a	7.6	9.1	1.9	
3	b	5.6	6.4	2.6	
4	c	14	1.2	3.4	
5	d	6.8	5.7	0.8	
6	e	18.5	17.7	5.3	
7	f	7.2	6.4	1.5	
8	g	19.5	16.6	4.5	
9	h	10.5	8.3	2.6	
10	i	5.27	4.9	1.16	
11	j	6	1	1.3	
12	k	8	1	2	
13	l	7.5	3	2	
14	m	23	23	6.7	
15	n	5.8	6.13	1.44	
16	o	11	9	2	
17	p	9	2	2	
18	q	6	4	1	
19	r	6	4.5	1	
20	s	9.5	10	1.5	
21	t	8	4	2	
22	u	25.6	27.2	4.1	
23	v	19	17	4.5	
24	w	9	9.8	3.4	
25	x	6.8	6.8	1.1	
26					
27					

# Another method

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	color	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x
2	blue	7.6	5.6	14	6.8	18.5	7.2	19.5	10.5	5.27	6	8	7.5	23	5.8	11	9	6	6	9.5	8	25.6	19	9	6.8
3	green	9.1	6.4	1.2	5.7	17.7	6.4	16.6	8.3	4.9	1	1	3	23	6.13	9	2	4	4.5	10	4	27.2	17	9.8	6.8
4	red	1.9	2.6	3.4	0.8	5.3	1.5	4.5	2.6	1.16	1.3	2	2	6.7	1.44	2	2	1	1	1.5	2	4.1	4.5	3.4	1.1
5																									
6																									

- columns for roach measured
- rows for color
- Also logical, although probably less convenient

# Tidy Data (tidyr)





# "Tidy data" is a buzzword

- Made up by Hadley Wickham, R guru.
- Rules:
  - Each variable must have its own column.
  - Each observation must have its own row.
  - Each value must have its own cell.
- See <https://r4ds.had.co.nz/tidy-data.html>

# What are the variables in the roach experiment?

- **block** and **color** are factors (discontinuous independent **variables**)
- **response** is a continuous dependent **variable**

**observations**

- So block, color, and response should be in separate columns if data are tidy.

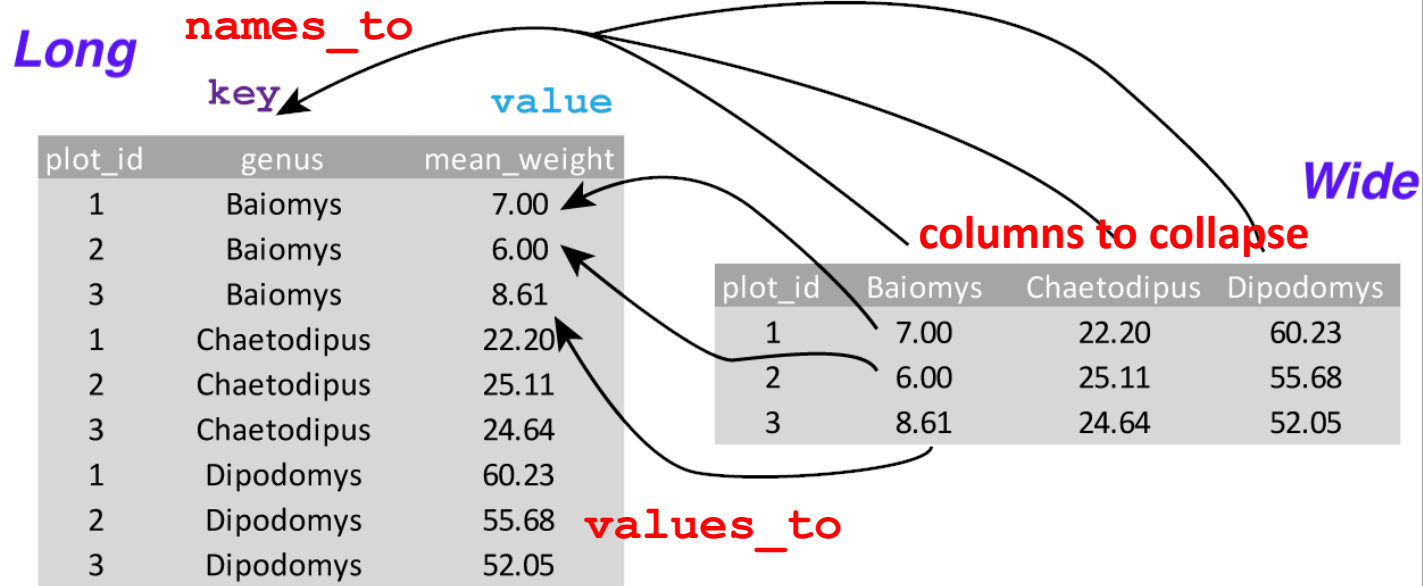
**variables**

	A	B	C	
1	block	color	response	
2	a	blue	7.6	
3	a	green	9.1	
4	a	red	1.9	
5	b	blue	5.6	
6	b	green	6.4	
7	b	red	2.6	
8	c	blue	14	
9	c	green	1.2	
10	c	red	2.4	
11	d	blue	6.8	
12	d	green	5.7	
13	d	red	0.8	
14	e	blue	18.5	
15	e	green	17.7	
16	e	red	5.3	
17	f	blue	7.2	
18	f	green	6.4	
19	f	red	1.5	
20	g	blue	19.5	
21	a	green	16.6	

# Pre-buzzword

- This format has been required by stats software for many years.
- Organizing factors in columns rather than mixing them in rows and columns makes them "**grouping variables**", since the software can use those columns to group the data in various ways
- "Tidy data" is a handy term for this format, so we'll use it.

# "Tidying" with tidyr: `pivot_longer()`



- "tidy" form = "long", "notebook" form = "wide"
- **key** = column to form from headers, **value** = data

```
pivot_longer(old_tibble_name,  
             cols = c("collapse_column1", "collapse_column2"),  
             names_to = "new_category_column",  
             values_to = "new_data_values_column")
```

# Examples with ERG data

# Untidying data

- One can use the **pivot\_wider()** function to reverse the tidying process.
- Result not good for **analysis** purposes, but sometimes easier for **data entry**.

# Modifying tibbles (dplyr)



# dplyr functions

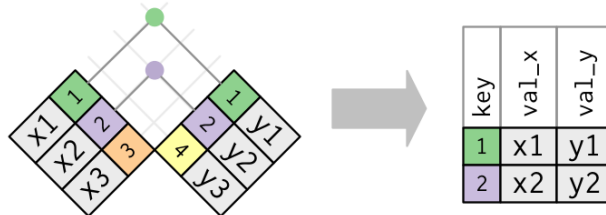
- **filter()** subsets rows
- **select()** subsets columns
- **mutate()** calculates new columns or changes existing ones



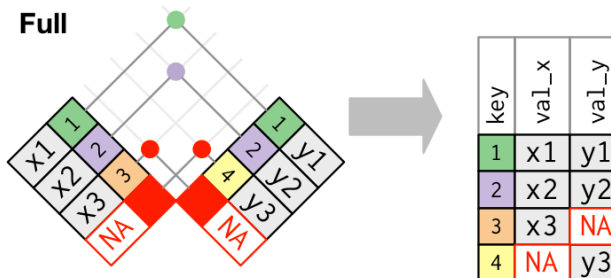
# Examples with schools data

# 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") )
```

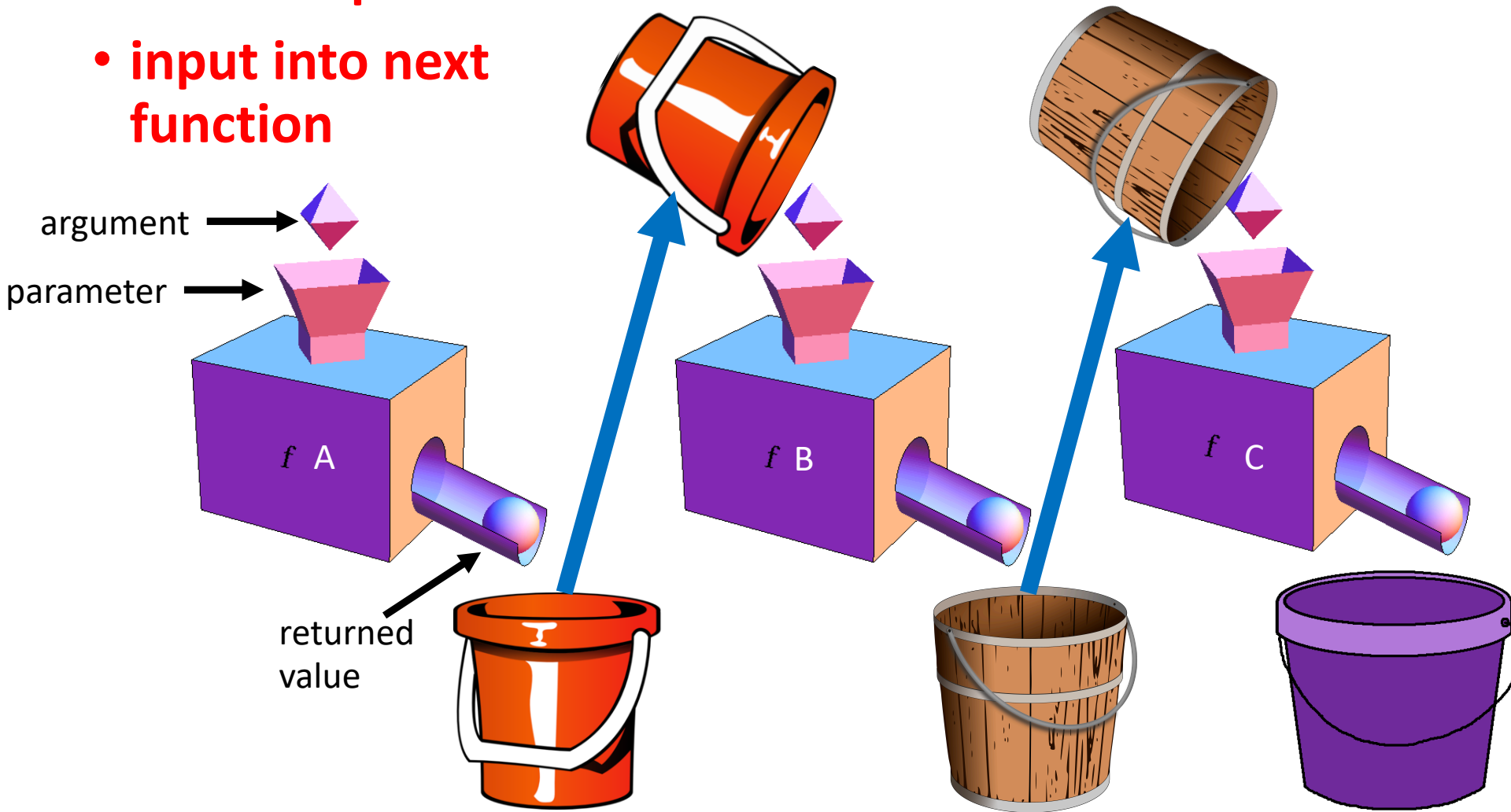
- First two arguments are the two tibbles to join
- **by** value are columns to join by; use = if names differ
- **suffix** value is added to columns with duplicate names
- other join types: **inner\_join()**, **left\_join()**, ...

pipelines (magrittr)



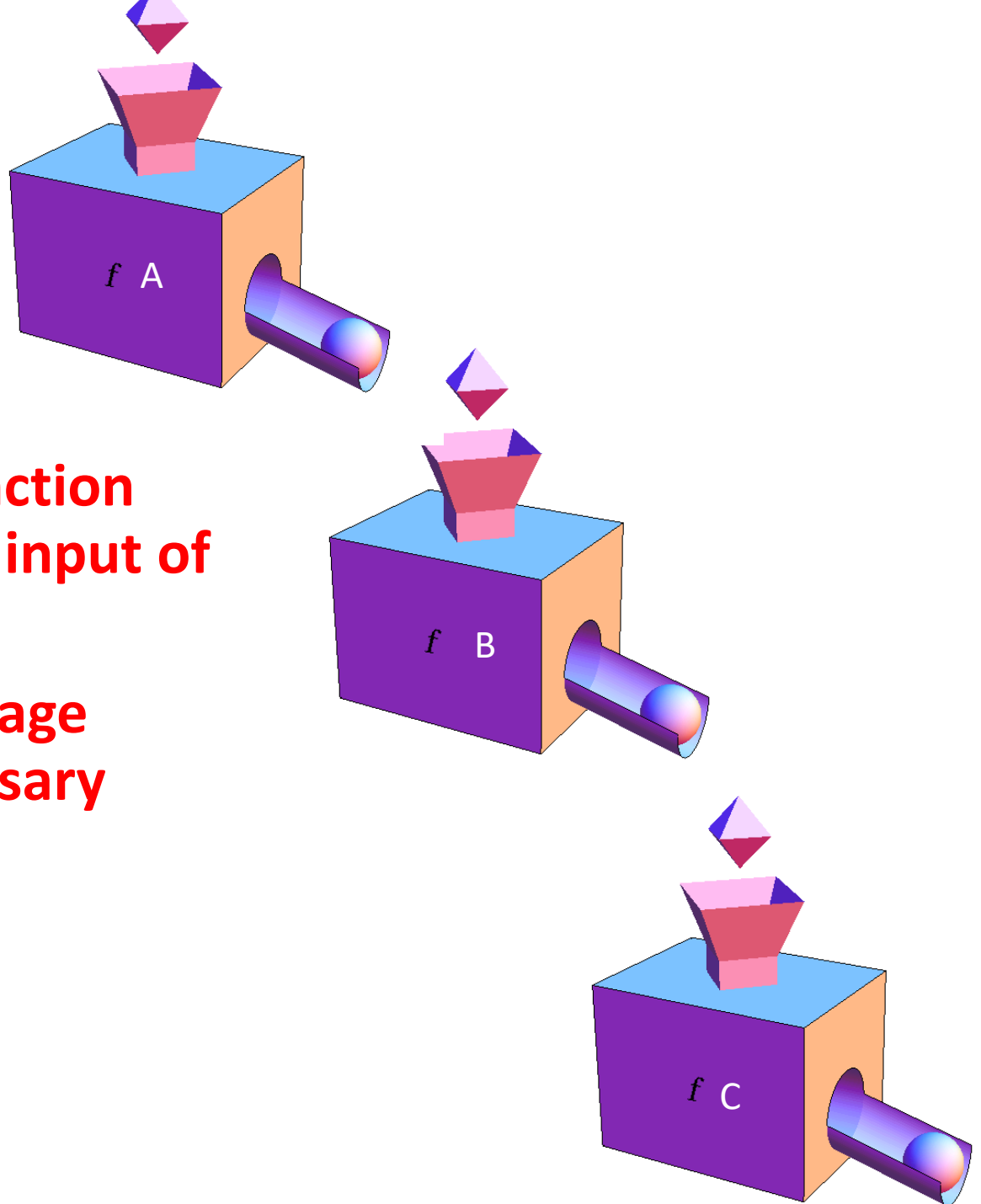
# Classic function/variable interaction

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



# Piping

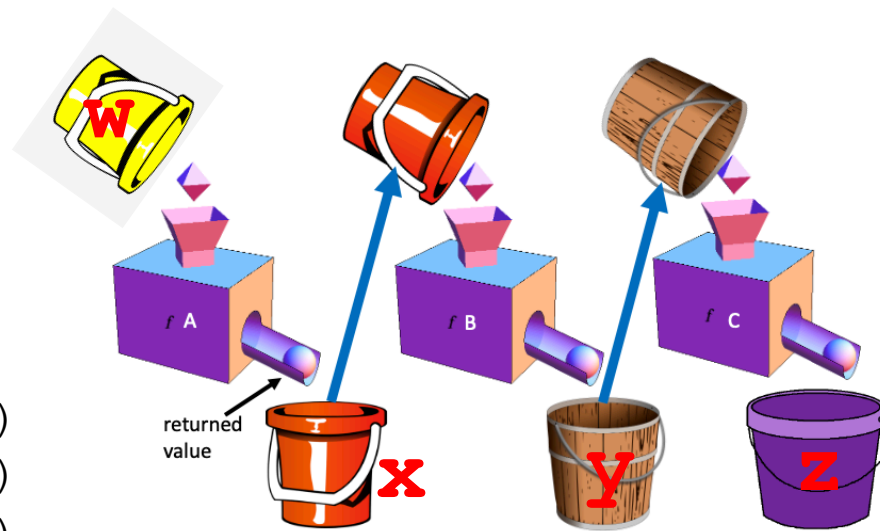
- output of one function goes directly into input of next
- intermediate storage objects not necessary



# Examples

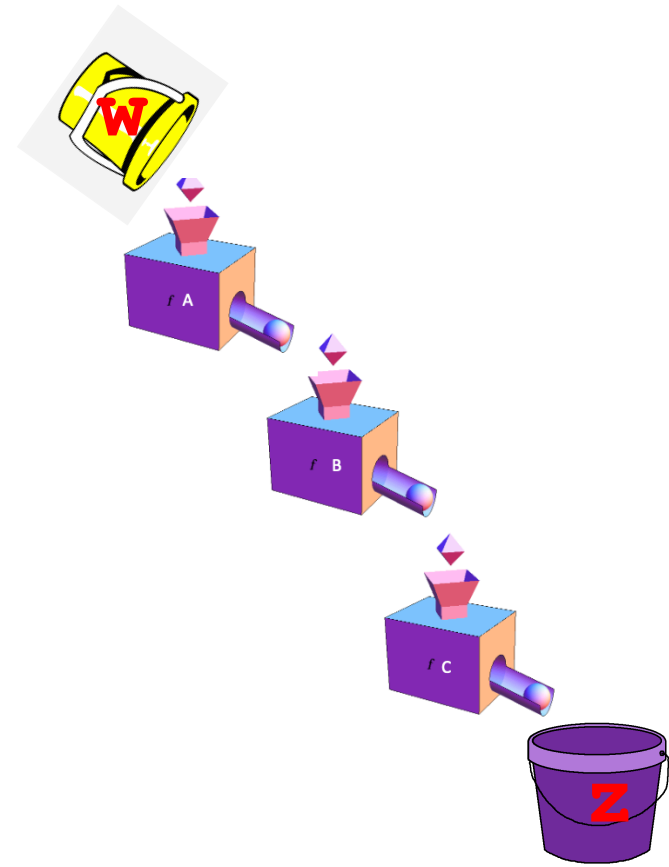
- Classic

```
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