Data Carpentry

Day 2

Spreadsheets

- Make it a rectangle
- Rows = observations, columns = variables
- One head row; avoid spaces
- One data type per cell
- Fill in all cells
- Consistently code missing values
- Care about date data
- Don’t do calculations in raw data files
- Save as CSV files
- Don’t use font color or highlighting to code data

OpenRefine

- For cleaning and exploration of data
- NOT for editing your raw data!
- Use Facets and filters to explore
- Split columns
- Remove training/ending text
- Find outliers
- All actions are reproducible

SQL

- SELECT (choose columns)
- FROM (data sheet(s))
- WHERE (subset specific observations)
- AND/OR/IN (used in setting criteria)
- ORDER BY (sort data)
- GROUP BY (lump data into groups)
- COUNT & SUM (summarization)
- JOIN ON (combining data)

dplyr

R function	SQL Keyword
select	SELECT
filter	WHERE
mutate	(weight/1000)
group_by	GROUP BY
summarize	COUNT, AVG, SUM
arrange	ORDER BY

"File organization and naming are powerful weapons against chaos."

-Jenny Bryan

Organizing projects

- All files in common folder (directory)
- Separate raw data from “clean” data
- Separate code (and output) from data
- Use file names that are meaningful, sortable, & consistent
- Code dates: 2017-01-11
    - raw_data/
    - in_process_data/
    - clean_data/
    - code/
    - reports/

“Your closest collaborator is you from six months ago, but you don’t reply to emails.”

-(Paraphrasing) Mark Holder

Have sympathy for your future self--be an organized analyst!

Today: R!

- Full programming language
- Focused on programming and data
- Super for data analysis and visualization
- Great community of supporters
- R Archive has >9000 add-on packages
- RStudio: “Integrated Development Environment” (IDE) for R

Challenge

What would *y* equal after these three lines of code were executed (try to answer without running them first!)? Why? How would you make it equal something else?

```
x <- 50
y <- x * 2
x <- 75
```

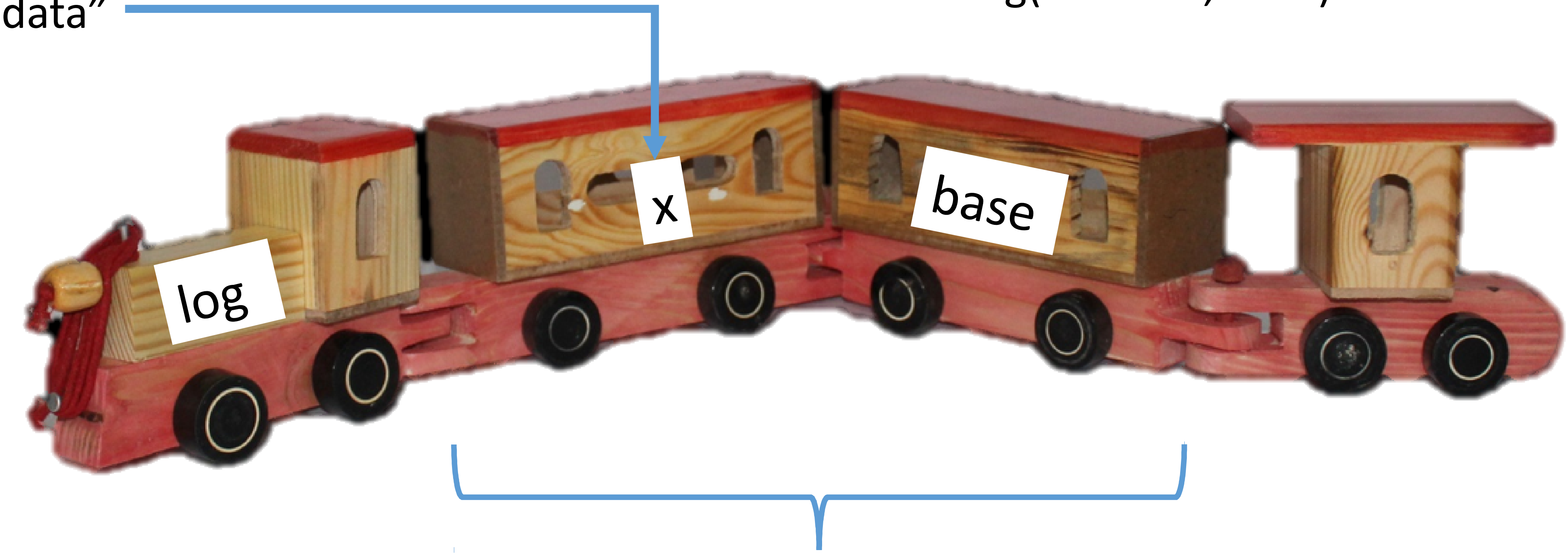
R Functions and inputs

log(x, base)

log(3, 5) = 0.683
log(5, 3) = 1.465

log(3, 5) = 0.683
log(base = 5, x = 3) = 0.683

x often = “data”



Arguments

R Objects

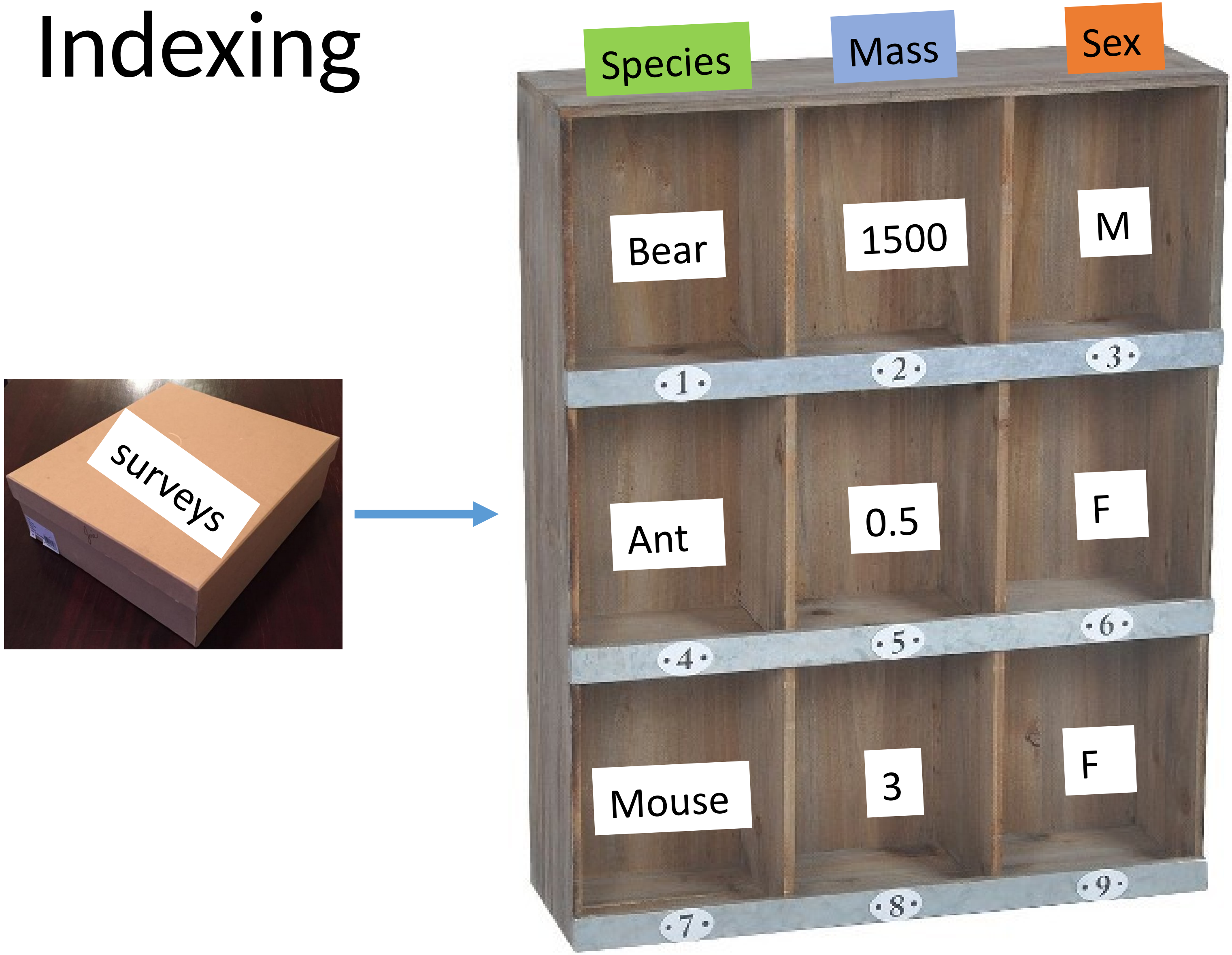
Stuff

Create object

Use name

Stuff

Indexing

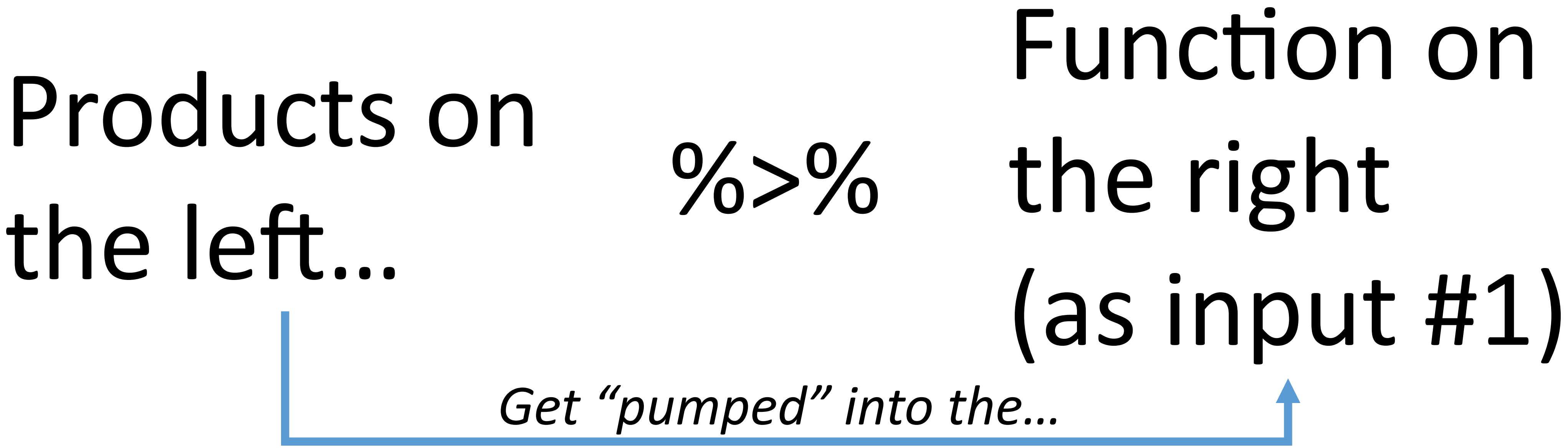


Species	Mass	Sex
Bear	1500	M
Ant	0.5	F
Mouse	3	F

Challenge

Use the *nrow*() function + indexing to save just the last row of *surveys* into a new object called *surveys_last*

Pipe Operator: %>%



Products on the left…

%>%

Function on the right (as input #1)

Get "pumped" into the…

You can string multiple operations together!