

R tidyverse data – a standard data format

“TIDY DATA” is a standard way of mapping the meaning of a dataset to its structure. ”

—HADLEY WICKHAM

In tidy data:

- each variable forms a column
- each observation forms a row
- each cell is a single measurement

each column a variable

each row an observation

id	name	color
1	floof	gray
2	max	black
3	cat	orange
4	donut	gray
5	merlin	black
6	panda	calico

Wickham, H. (2014). Tidy Data. Journal of Statistical Software 59 (10). DOI: 10.18637/jss.v059.i10

Filter data only to certain rows (observations)

dplyr::filter()

KEEP ROWS THAT
satisfy
your CONDITIONS

keep rows from... this data... ONLY IF... type is "otter" AND site is "bay"

```
filter(df, type == "otter" & site == "bay")
```



	type	food	site
1	otter	urchin	bay
2	Shark	seal	channel
3	otter	abalone	bay
4	otter	crab	wharf

@allison_horst

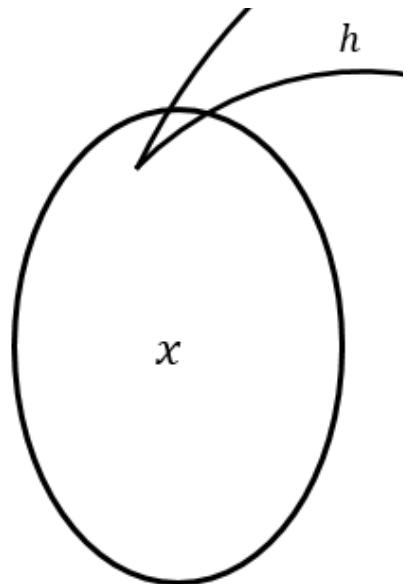
Adding new columns (variables) – or performing functions on the ones you have



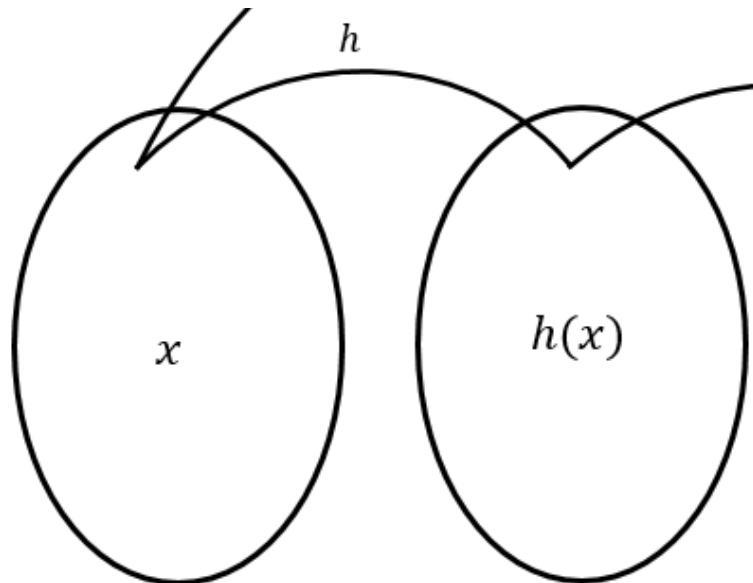
And many other useful commands

- `slice()`: pick rows using index(es)
- `filter()`: pick rows matching criteria
- `select()`: pick columns by name or index
- `pull()`: grab a column as a vector
- `rename()`: rename specific columns
- `arrange()`: reorder rows
- `mutate()`: add new variables
- `transmute()`: create new data frame with variables
- `group_by()`: create groups inside the dataset
- `summarise()`: reduce variables to values
- `distinct()`: filter for unique rows

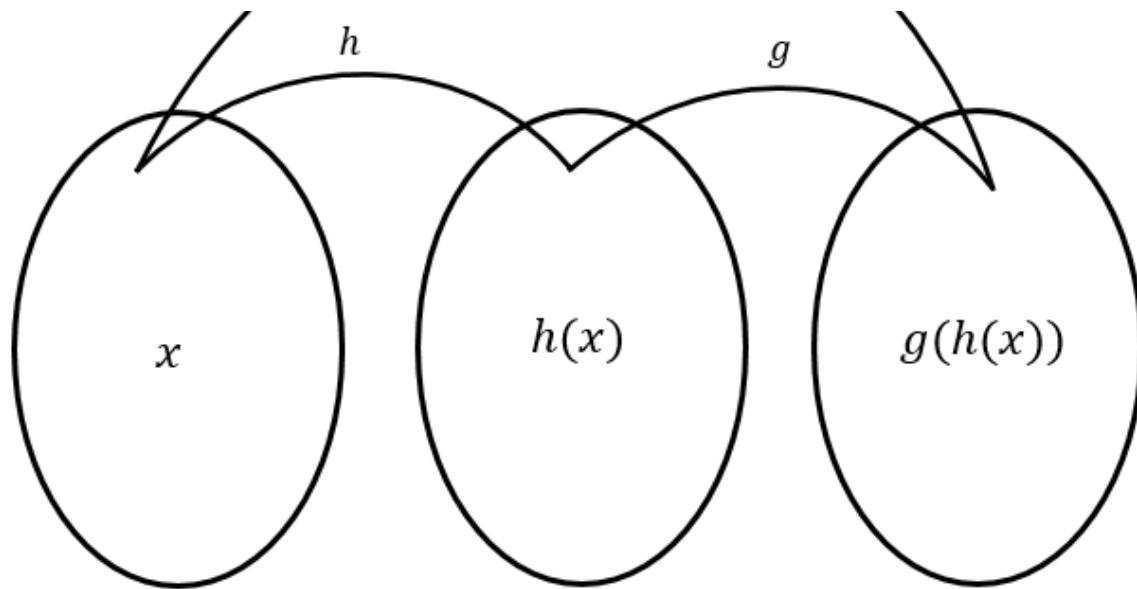
Pipes in R



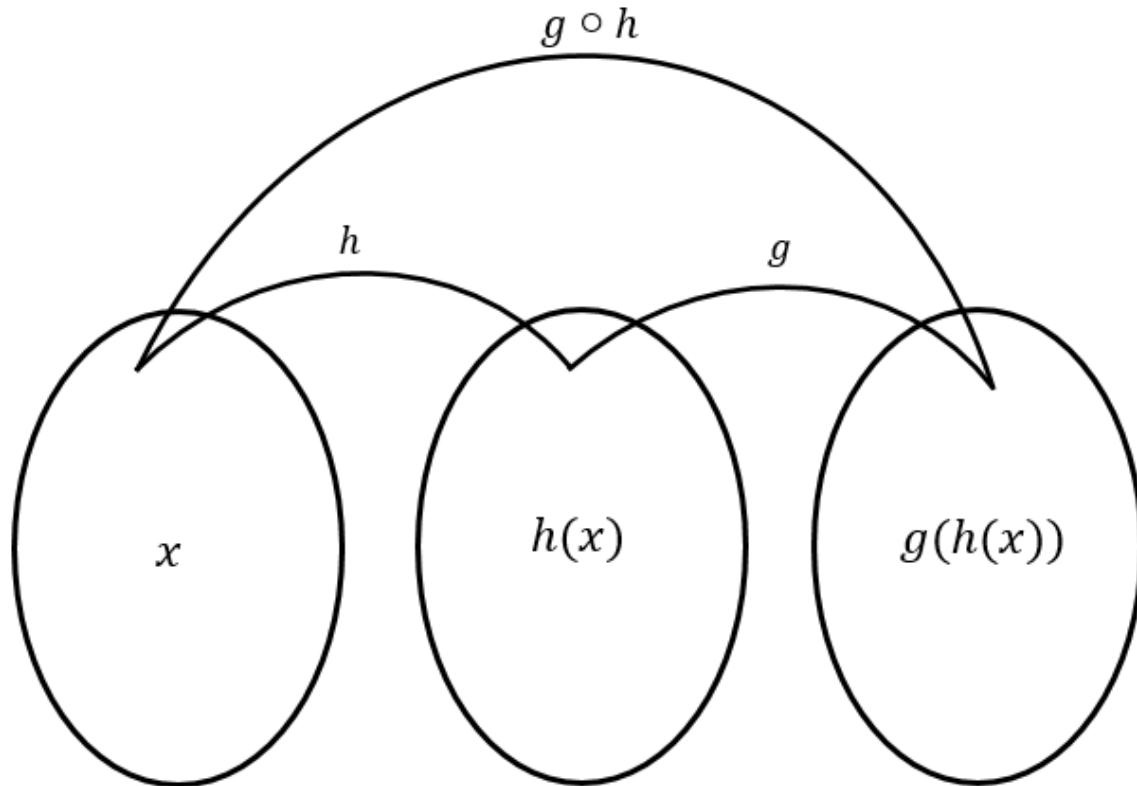
Pipes in R



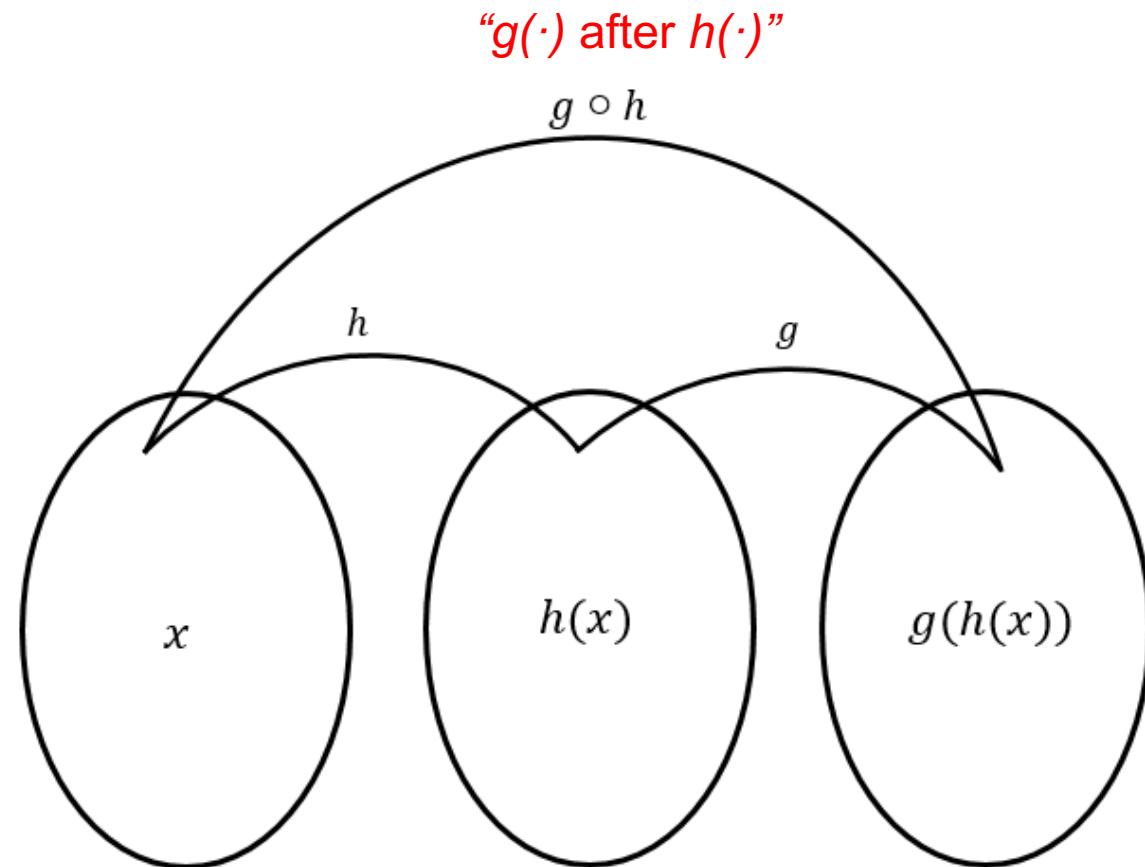
Pipes in R



Pipes in R



Pipes in R



Pipes in R

```
# Initialize `x`  
> x <- c(0.109, 0.359, 0.63, 0.996, 0.515, 0.142, 0.017, 0.829,  
0.907)
```

```
# Compute the logarithm of `x`, return lagged and iterated #  
differences, compute the exponential function and round # the  
result  
> round(  
  exp(  
    diff(  
      log(x)  
    ))  
, 1  
)
```

Pipes in R

```
# Without pipes
```

```
> round(exp(diff(log(x))), 1)
```

```
# Same code, with pipes
```

```
> x %>%  
  log() %>%  
  diff() %>%  
  exp() %>%  
  round(1)
```

“log then diff then exp then round”

We will learn a lot more as we go, through every workshop