Tables, Recodes, Regexps

~/> previously …

Reasonable Grad Students:
We want practical data skills applicable both in research and outside of academia.

Me: OK, here is git and how to use it.

Reasonable Grad Students:

The name of the function, and the library it is in.

mean {base} R Documentation

Arithmetic Mean

Description

What it does.

Generic function for the (trimmed) arithmetic mean.

Usage

```
mean(x, ...)

## Default S3 method:
mean(x, trim = 0, na.rm = FALSE, ...)
```

The function's name, and in the parentheses the named arguments it expects, in the order it expects them. If an argument has a default value, it is shown. Arguments without default values (e.g. x) must be provided by you.

Arguments

More details on each named argument. This will tell you what class of thing each argument has to be—an object, a number, a data frame, a logical value, etc.

`x` An R object. Currently there are methods for numeric/logical vectors and date, date-time and time interval objects. Complex vectors are allowed for `trim = 0`, only.

`trim` the fraction (0 to 0.5) of observations to be trimmed from each end of `x` before the mean is computed. Values of trim outside that range are taken as the nearest endpoint.

`na.rm` a logical value indicating whether `NA` values should be stripped before the computation proceeds.

`...` further arguments passed to or from other methods.

The ellipsis allows other arguments to be passed to and from the function.

Value

What the function returns—i.e., the result of whatever operation or calculation it performs. This can be a single number, as here, or a multi-part object such as a list, a data frame, a plot, or a model.

If `trim` is zero (the default), the arithmetic mean of the values in `x` is computed, as a numeric or complex vector of length one. If x is not logical (coerced to numeric), numeric (including integer) or complex, `NA_real_` is returned, with a warning.

If `trim` is non-zero, a symmetrically trimmed mean is computed with a fraction of `trim` observations deleted from each end before the mean is computed.

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

See Also

`weighted.mean`, `mean.POSIXct`, `colMeans` for row and column means.

Other related functions

Examples

```
x <- c(0:10, 50)
xm <- mean(x)
c(xm, mean(x, trim = 0.10))
```

Self-contained examples that you can run at the console. These may use built-in datasets or other R functions.

[Package *base* version 3.4.3 Index]

Visit the package's Index page to look for Demos and Vignettes detailing how it works.

~/>_

WORKING WITH DPLYR

```r
data <- data %>%
  mutate(poc = recode(race, "White" = "White", .default = "Non-White"),
         days_old = calc_age(born, start_year, "day"),
         months_old = calc_age(born, start_year, "month"),
         full_name = paste(first, last, suffix),
         full_name = stringr::str_remove(full_name, " NA$"),
         entry_age = calc_age(born, start),
         yr_fac = factor(year(start_year)))
```

```r
median_age_party <- data %>%
  filter(position == "U.S. Representative") %>%
  group_by(congress, party) %>%
    summarize(year = first(start_year),
              median_age = median(start_age)) %>%
    filter(party %in% c("Democrat", "Republican"))

median_age_party
```

```r
oldest_group_by_year <- data %>%
  filter(party %in% c("Democrat", "Republican"),
         position == "U.S. Representative") %>%
  group_by(congress, party) %>%
  filter(start_age > quantile(start_age, 0.99, na.rm = TRUE))

youngest_group_by_year <- data %>%
  filter(party %in% c("Democrat", "Republican"),
         position == "U.S. Representative") %>%
  group_by(congress, party) %>%
  filter(start_age < quantile(start_age, 0.01, na.rm = TRUE))
```

```
data %>%
    select(start_year, job_type1) %>%
    group_by(start_year, job_type1) %>%
    summarize(n = n()) %>%
    mutate(pct = (n/sum(n))*100)
```

```
data %>%
    select(start_year, job_type1) %>%
    group_by(start_year, job_type1) %>%
    summarize(n = n()) %>%
    mutate(pct = (n/sum(n))*100) %>%
    group_by(start_year) %>%
    top_n(3, wt = pct)
```

```r
data %>%
    select(start_year, job_type1) %>%
    group_by(start_year, job_type1) %>%
    summarize(n = n()) %>%
    mutate(pct = (n/sum(n))*100) %>%
    group_by(start_year) %>%
    top_n(3, wt = pct)
```

```r
data %>%
    select(start_year, job_type1) %>%
    group_by(start_year, job_type1) %>%
    summarize(n = n()) %>%
    mutate(pct = (n/sum(n))*100) %>%
    group_by(start_year) %>%
    top_n(3, wt = pct) %>%
    arrange(desc(pct))
```

```
data %>%
    select(start_year, job_type1) %>%
    group_by(start_year, job_type1) %>%
    summarize(n = n()) %>%
    mutate(pct = (n/sum(n))*100) %>%
    group_by(start_year) %>%
    top_n(3, wt = pct) %>%
    arrange(desc(pct), .by_group = TRUE)
```

~/>_

TIDY DATA

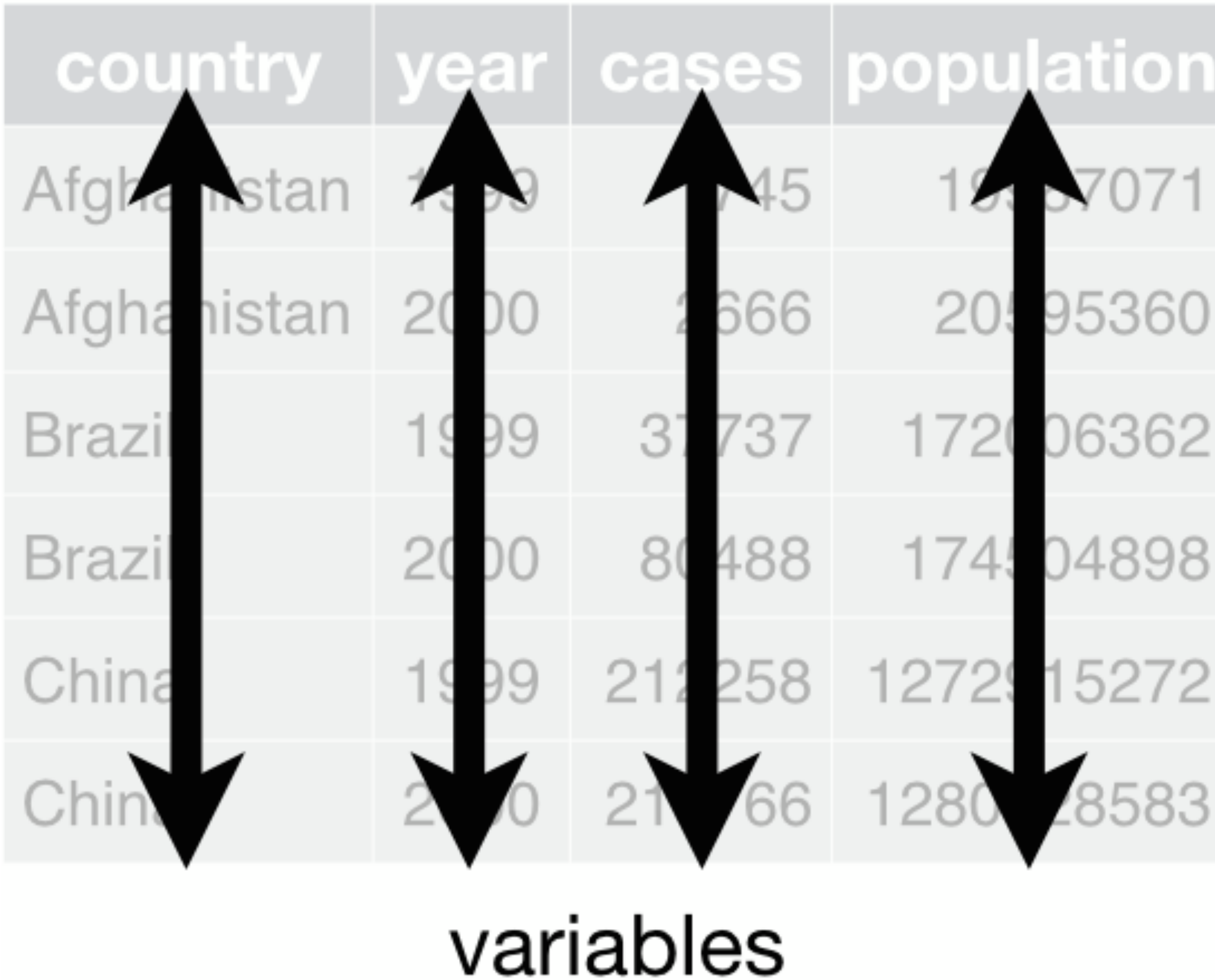
MOST DATA ANALYSIS IS CLEANING & RECODING

```
library(socviz)

library(gapminder)

gapminder
```

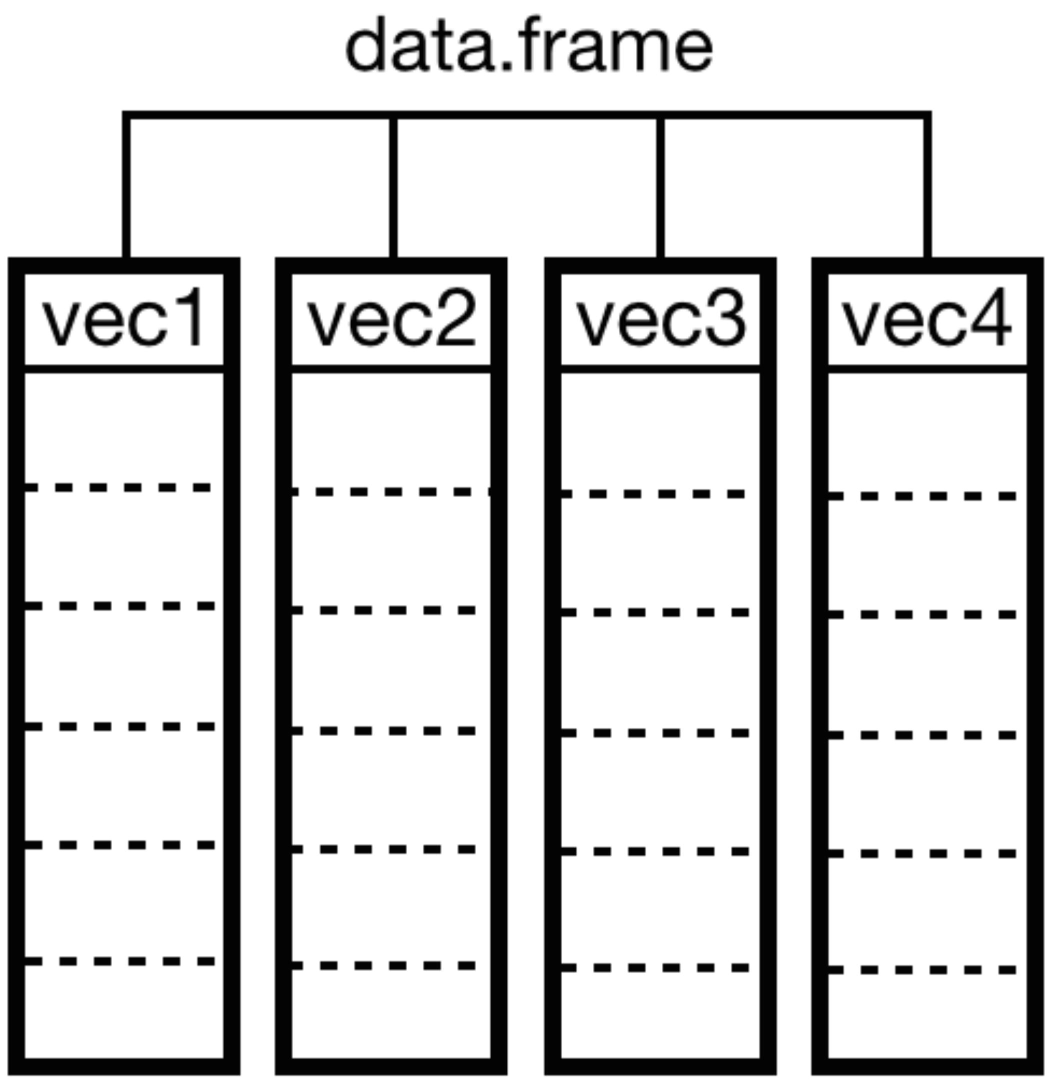
gdp	lifexp	pop	continent
340	65	31	Euro
227	51	200	Amer
909	81	80	Euro
126	40	20	Asia


country
year
cases
population
variables

observations

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

values


data.frame
vec1
vec2
vec3
vec4

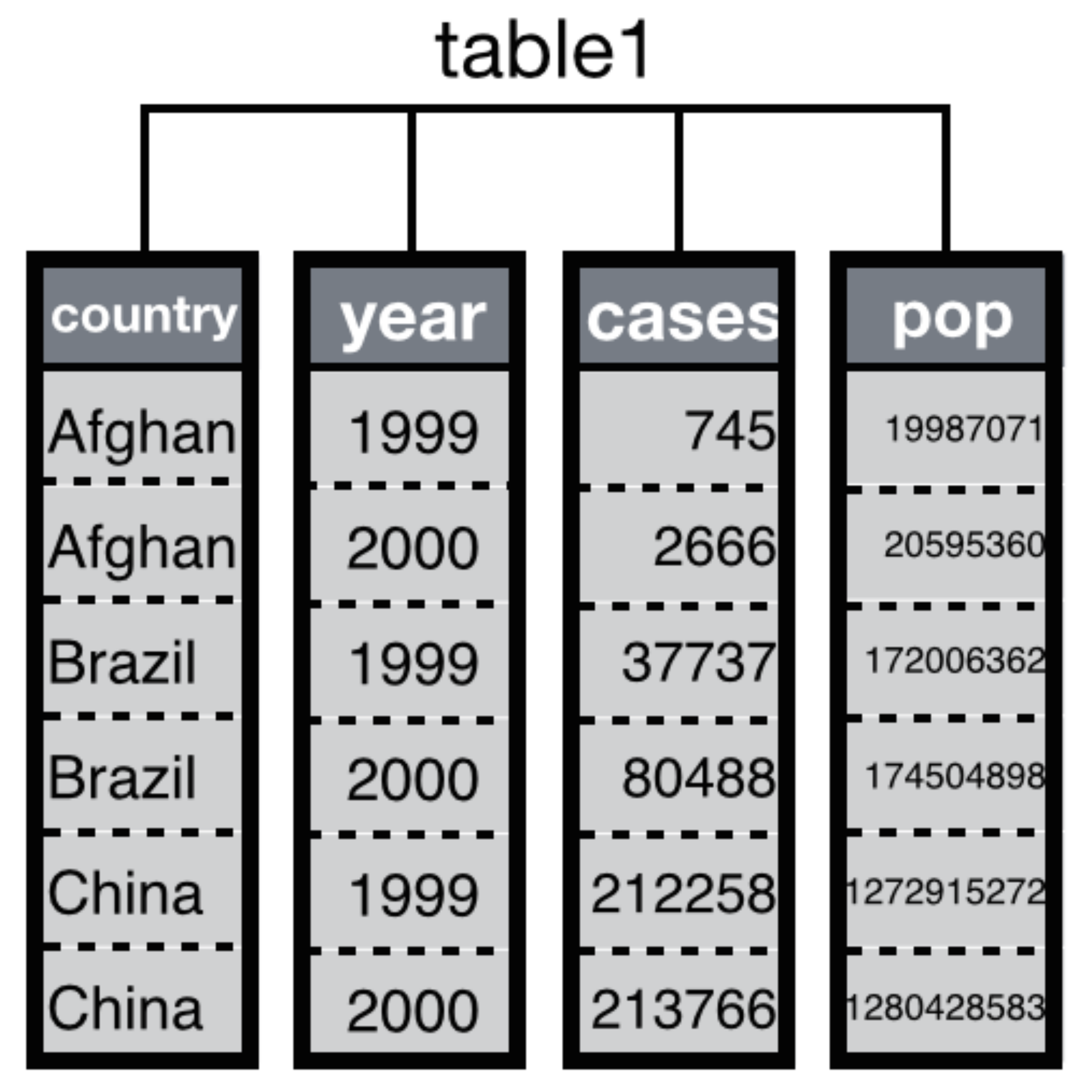


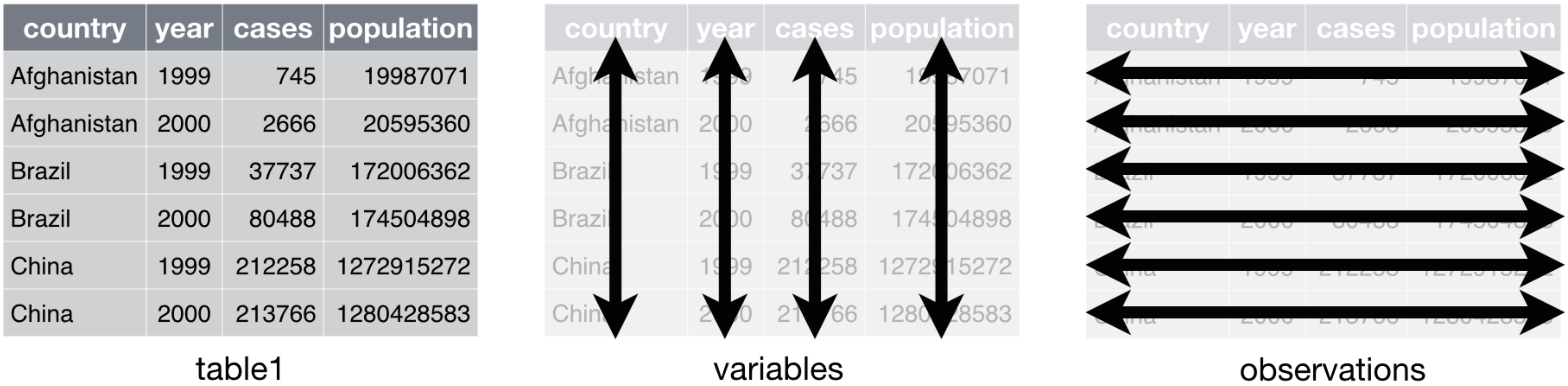
table1
country
year
cases
pop
Afghan
1999
745
19987071
Afghan
2000
2666
20595360
Brazil
1999
37737
172006362
Brazil
2000
80488
174504898
China
1999
212258
1272915272
China
2000
213766
1280428583

State

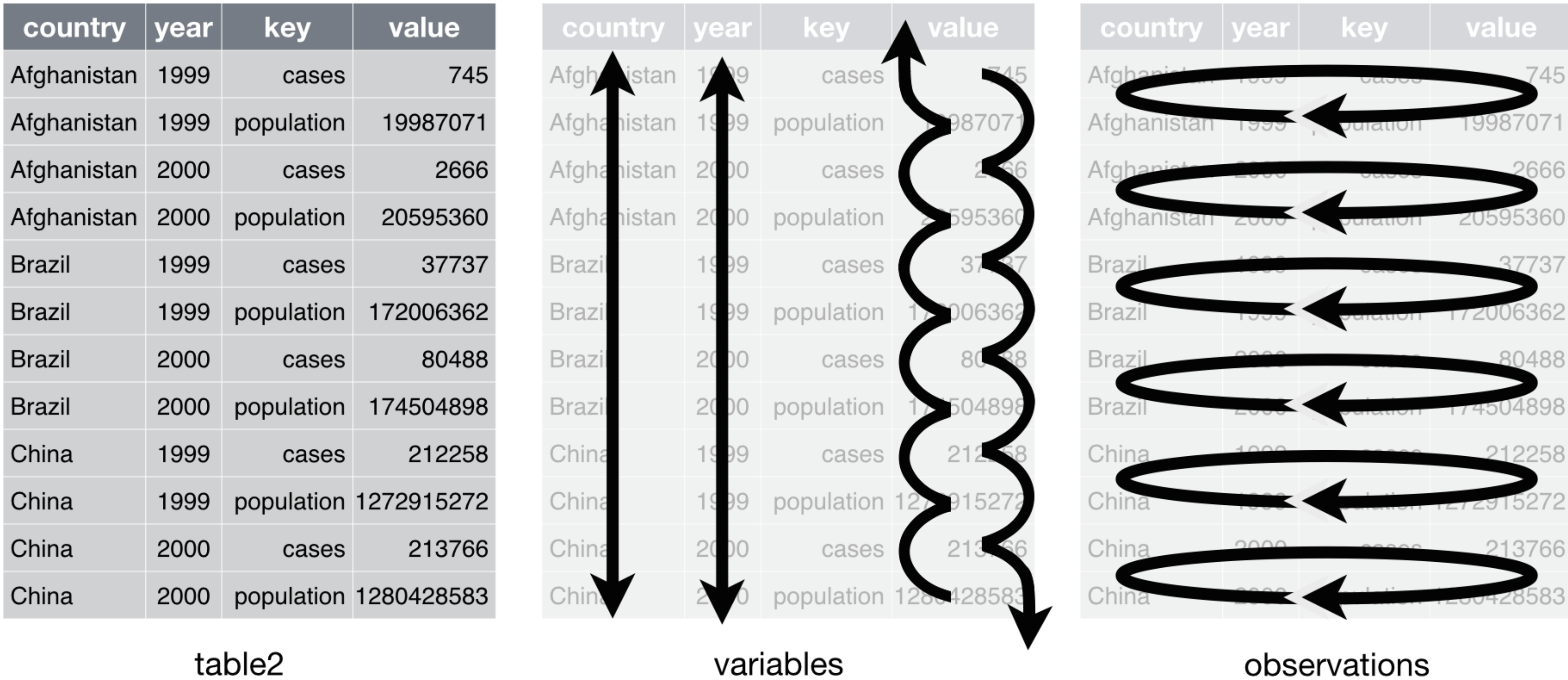
A	B	C	D	E	F	G	H	I	J	K	L	M	N	P	Q
State	**CD#**	**2018 Cook PVI Score**	**2018 Winner**	**Party**	**Dem Votes**	**GOP Votes**	**Other Votes**	**Dem %**	**GOP %**	**Other %**	**Dem Margin**	**2016 Clinton Margin**	**Swing vs. 2016 Prez**	**Raw Votes vs. 2016**	**Final?**
New House Breakdown: 235D, 199R, 1 Not Certified				**D**	**60,619,428**	**50,896,244**	**1,978,795**	**53.4%**	**44.8%**	**1.7%**	**8.6%**	**2.1%**	**6.5%**	**83.3%**	
Compiled by: David Wasserman & Ally Flinn, Cook Political Report. @Redistrict/@CookPolitical. *Italics* denotes freshman, **Bold** denotes party change.															
Alabama	1	R+15	Bradley Byrne	R	89,226	153,228	163	36.8%	63.2%	0.1%	-26.4%	-29.2%	2.8%	79.3%	x
Alabama	2	R+16	Martha Roby	R	86,931	138,879	420	38.4%	61.4%	0.2%	-23.0%	-31.7%	8.7%	78.7%	x
Alabama	3	R+16	Mike Rogers	R	83,996	147,770	149	36.2%	63.7%	0.1%	-27.5%	-33.0%	5.5%	79.6%	x
Alabama	4	R+30	Robert Aderholt	R	46,492	184,255	222	20.1%	79.8%	0.1%	-59.6%	-62.5%	2.9%	78.9%	x
Alabama	5	R+18	Mo Brooks	R	101,388	159,063	222	38.9%	61.0%	0.1%	-22.1%	-32.9%	10.8%	82.8%	x
Alabama	6	R+26	Gary Palmer	R	85,644	192,542	142	30.8%	69.2%	0.1%	-38.4%	-43.8%	5.4%	82.8%	x
Alabama	7	D+20	Terri Sewell	D	185,010	0	4,153	97.8%	0.0%	2.2%	97.8%	41.2%	N/A	64.2%	x
Alaska	AL	R+9	Don Young	R	131,199	149,779	1,188	46.5%	53.1%	0.4%	-6.6%	-14.7%	8.1%	88.6%	x
Arizona	1	R+2	Tom O'Halleran	D	143,240	122,784	65	53.8%	46.1%	0.0%	7.7%	-1.1%	8.8%	92.0%	x
Arizona	2	R+1	***Ann Kirkpatrick***	***D***	161,000	133,102	50	54.7%	45.2%	0.0%	9.5%	4.8%	4.7%	91.5%	x
Arizona	3	D+13	Raul Grijalva	D	114,650	64,868	0	63.9%	36.1%	0.0%	27.7%	29.5%	-1.8%	84.8%	x
Arizona	4	R+21	Paul Gosar	R	84,521	188,842	3,672	30.5%	68.2%	1.3%	-37.7%	-39.4%	1.7%	91.1%	x
Arizona	5	R+15	Andy Biggs	R	127,027	186,037	0	40.6%	59.4%	0.0%	-18.8%	-20.5%	1.7%	91.7%	x
Arizona	6	R+9	David Schweikert	R	140,559	173,140	0	44.8%	55.2%	0.0%	-10.4%	-9.8%	-0.6%	91.2%	x
Arizona	7	D+23	Ruben Gallego	D	113,044	301	18,706	85.6%	0.2%	14.2%	85.4%	48.3%	N/A	79.0%	x
Arizona	8	R+13	Debbie Lesko	R	135,569	168,835	13	44.5%	55.5%	0.0%	-10.9%	-20.8%	9.9%	91.5%	x
Arizona	9	D+4	*Greg Stanton*	*D*	159,583	101,662	0	61.1%	38.9%	0.0%	22.2%	15.9%	6.3%	90.0%	x
Arkansas	1	R+17	Rick Crawford	R	57,907	138,757	4,581	28.8%	68.9%	2.3%	-40.2%	-34.8%	-5.4%	77.2%	x
Arkansas	2	R+7	French Hill	R	116,135	132,125	5,193	45.8%	52.1%	2.0%	-6.3%	-10.7%	4.4%	82.6%	x
Arkansas	3	R+19	Steve Womack	R	74,952	148,717	6,039	32.6%	64.7%	2.6%	-32.1%	-31.4%	-0.7%	78.6%	x
Arkansas	4	R+17	Bruce Westerman	R	63,984	136,740	4,168	31.2%	66.7%	2.0%	-35.5%	-32.8%	-2.7%	75.7%	x
California	1	R+11	Doug LaMalfa	R	131,506	160,006	0	45.1%	54.9%	0.0%	-9.8%	-19.4%	9.6%	91.6%	
California	2	D+22	Jared Huffman	D	243,051	72,541	0	77.0%	23.0%	0.0%	54.0%	45.2%	8.8%	90.5%	
California	3	D+5	John Garamendi	D	132,983	96,106	0	58.0%	42.0%	0.0%	16.1%	12.5%	3.6%	86.8%	
California	4	R+10	Tom McClintock	R	156,253	184,401	0	45.9%	54.1%	0.0%	-8.3%	-14.5%	6.2%	94.6%	
California	5	D+21	Mike Thompson	D	203,012	0	53,836	79.0%	0.0%	21.0%	79.0%	44.6%	N/A	83.8%	
California	6	D+21	Doris Matsui	D	201,939	0	0	100.0%	0.0%	0.0%	100.0%	44.0%	N/A	81.4%	
California	7	D+3	Ami Bera	D	155,016	126,601	0	55.0%	45.0%	0.0%	10.1%	11.2%	-1.1%	91.0%	
California	8	R+9	Paul Cook	R	0	170,785	0	0.0%	100.0%	0.0%	-100.0%	-15.1%	N/A	73.3%	
California	9	D+8	Jerry McNerney	D	113,240	87,263	0	56.5%	43.5%	0.0%	13.0%	18.2%	-5.2%	82.4%	

readxl

part of the tidyverse



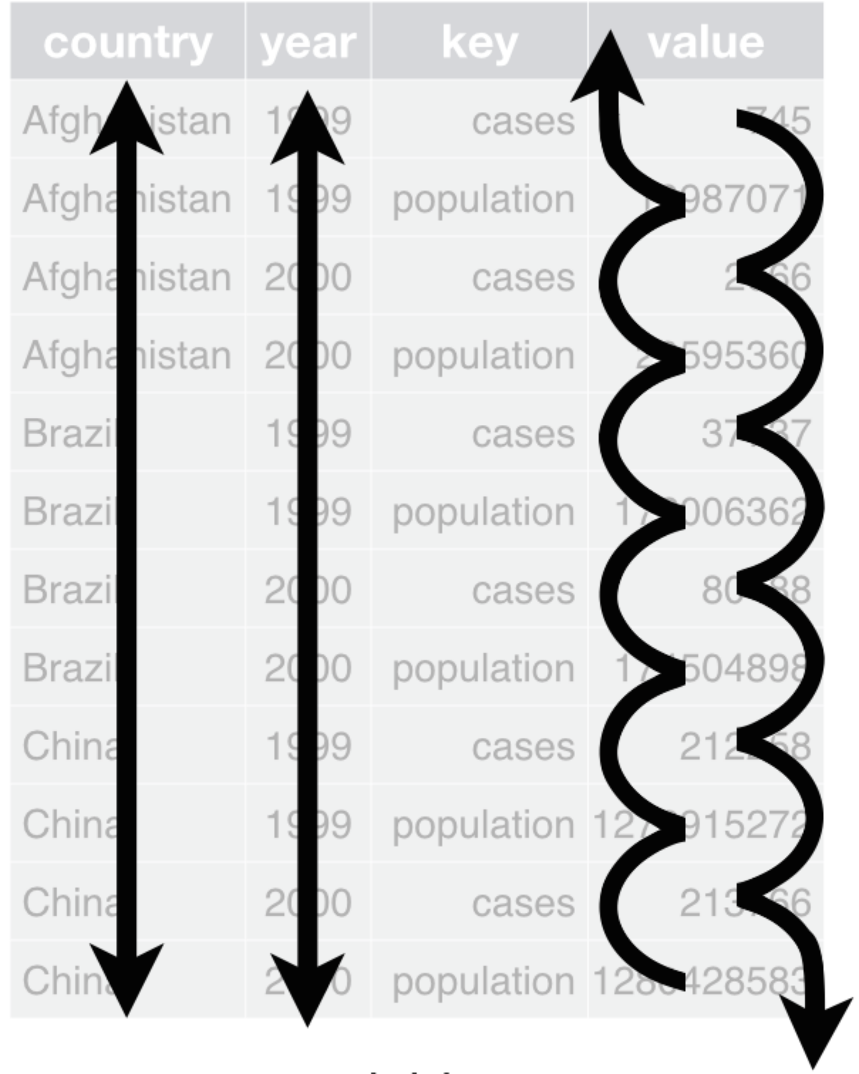

```
      country year  cases population
1 Afghanistan 1999    745   19987071
2 Afghanistan 2000   2666   20595360
3      Brazil 1999  37737  172006362
4      Brazil 2000  80488  174504898
5       China 1999 212258 1272915272
6       China 2000 213766 1280428583
```



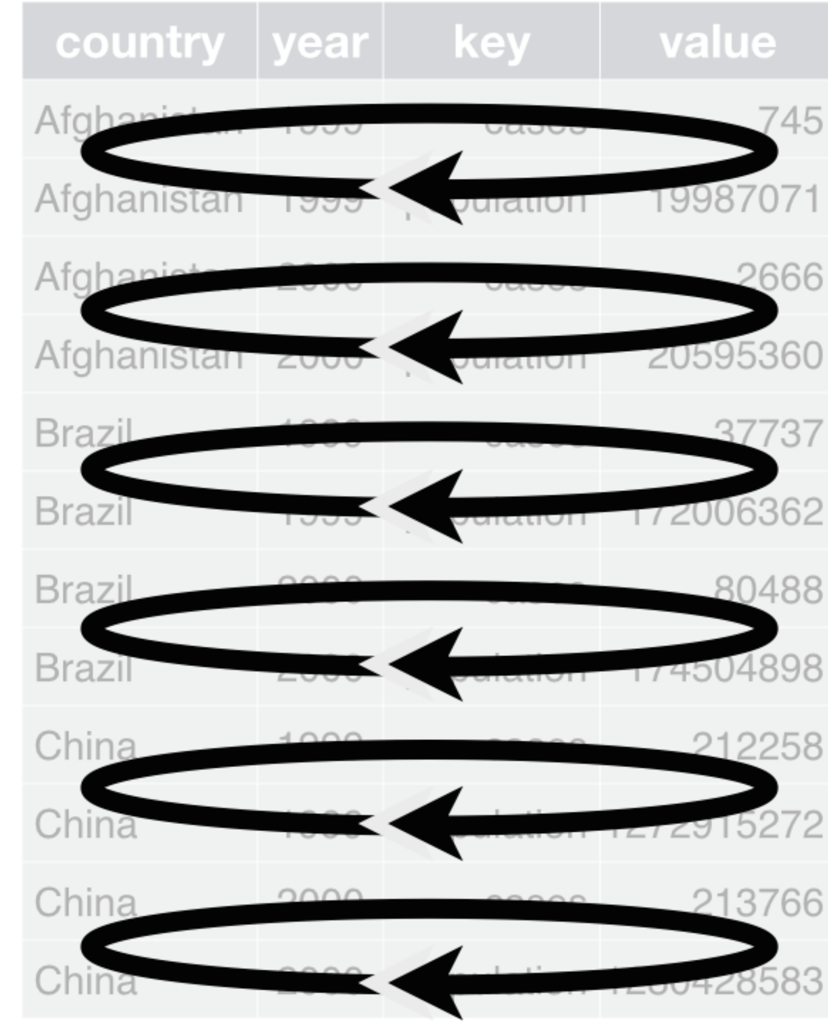

```
      country year        key      value
1  Afghanistan 1999      cases        745
2  Afghanistan 1999 population   19987071
3  Afghanistan 2000      cases       2666
4  Afghanistan 2000 population   20595360
5       Brazil 1999      cases      37737
6       Brazil 1999 population  172006362
7       Brazil 2000      cases      80488
8       Brazil 2000 population  174504898
9        China 1999      cases     212258
10       China 1999 population 1272915272
11       China 2000      cases     213766
12       China 2000 population 1280428583
```

country	year	key	value
Afghanistan	1999	cases	745
Afghanistan	1999	population	19987071
Afghanistan	2000	cases	2666
Afghanistan	2000	population	20595360
Brazil	1999	cases	37737
Brazil	1999	population	172006362
Brazil	2000	cases	80488
Brazil	2000	population	174504898
China	1999	cases	212258
China	1999	population	1272915272
China	2000	cases	213766
China	2000	population	1280428583

table2

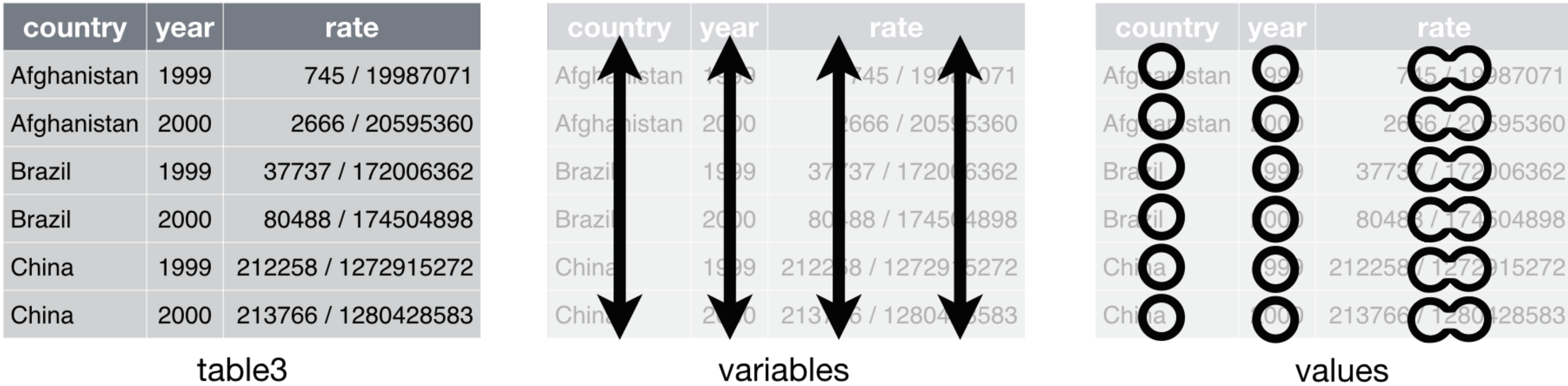


variables



observations

```
       country year        key      value
1  Afghanistan 1999      cases        745
2  Afghanistan 1999 population   19987071
3  Afghanistan 2000      cases       2666
4  Afghanistan 2000 population   20595360
5       Brazil 1999      cases      37737
6       Brazil 1999 population  172006362
7       Brazil 2000      cases      80488
8       Brazil 2000 population  174504898
9        China 1999      cases     212258
10       China 1999 population 1272915272
11       China 2000      cases     213766
12       China 2000 population 1280428583
```




```
       country year               rate
1 Afghanistan 1999      745/19987071
2 Afghanistan 2000     2666/20595360
3      Brazil 1999   37737/172006362
4      Brazil 2000   80488/174504898
5       China 1999 212258/1272915272
6       China 2000 213766/1280428583
```

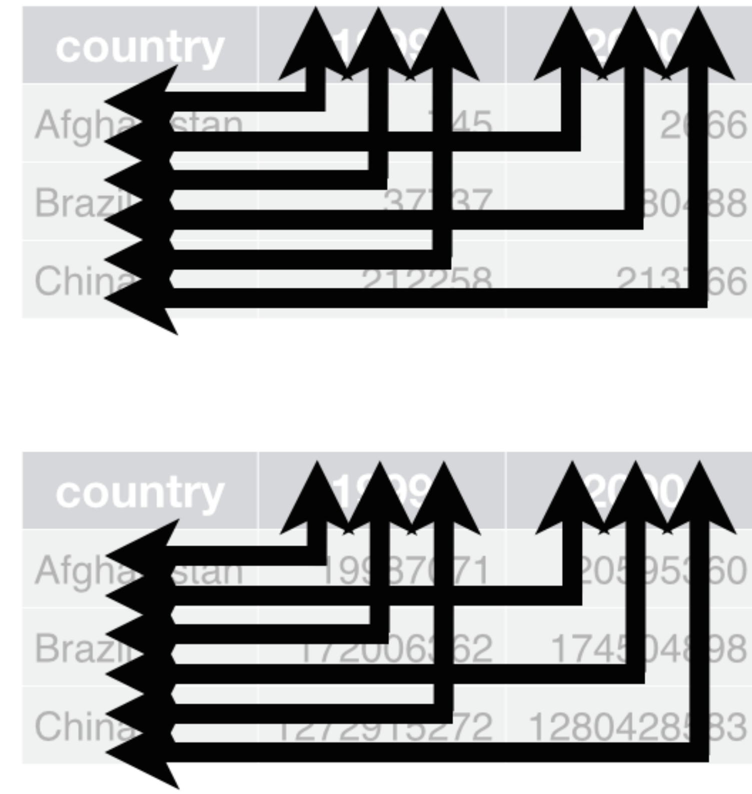
country	1999	2000
Afghanistan	745	2666
Brazil	37737	80488
China	212258	213766

table4

country	1999	2000
Afghanistan	19987071	20595360
Brazil	172006362	174504898
China	1272915272	1280428583

table5

variables



observations

```
      country   1999   2000
1 Afghanistan    745   2666
2      Brazil  37737  80488
3       China 212258 213766
```

```
      country       1999       2000
1 Afghanistan   19987071   20595360
2      Brazil  172006362  174504898
3       China 1272915272 1280428583
```

Table A-1. Years of School Completed by People 25 Years and Over, by Age and Sex: Selected Years 1940 to 2016

(Numbers in thousands. Noninstitutionalized population except where otherwise specified.)

Age, sex, and years	Total	Years of School Completed						
		Elementary		High school		College		Median
		0 to 4 years	5 to 8 years	1 to 3 years	4 years	1 to 3 years	4 years or more	
25 YEARS AND OLDER								
Male								
2016	103,372	1,183	3,513	7,144	30,780	26,468	34,283	(NA)
2015	101,887	1,243	3,669	7,278	30,997	25,778	32,923	(NA)
2014	100,592	1,184	3,761	7,403	30,718	25,430	32,095	(NA)
2013	99,305	1,127	3,836	7,314	30,014	25,283	31,731	(NA)
2012	98,119	1,237	3,879	7,388	30,216	24,632	30,766	(NA)
2011	97,220	1,234	3,883	7,443	30,370	24,319	29,971	(NA)
2010	96,325	1,279	3,931	7,705	30,682	23,570	29,158	(NA)
2009	95,518	1,372	4,027	7,754	30,025	23,634	28,706	(NA)
2008	94,470	1,310	4,136	7,853	29,491	23,247	28,433	(NA)
2007	93,421	1,458	4,249	8,294	29,604	22,219	27,596	(NA)
2006	92,233	1,472	4,395	7,940	29,380	22,136	26,910	(NA)
2005	90,899	1,505	4,402	7,787	29,151	21,794	26,259	(NA)

```
edu

## # A tibble: 366 x 11
##    age   sex    year total elem4 elem8   hs3   hs4 coll3 coll4 median
##    <chr> <chr> <int> <int> <int> <int> <dbl> <dbl> <dbl> <dbl>  <dbl>
##  1 25-34 Male   2016 21845   116   468  1427  6386  6015  7432     NA
##  2 25-34 Male   2015 21427   166   488  1584  6198  5920  7071     NA
##  3 25-34 Male   2014 21217   151   512  1611  6323  5910  6710     NA
##  4 25-34 Male   2013 20816   161   582  1747  6058  5749  6519     NA
##  5 25-34 Male   2012 20464   161   579  1707  6127  5619  6270     NA
##  6 25-34 Male   2011 20985   190   657  1791  6444  5750  6151     NA
##  7 25-34 Male   2010 20689   186   641  1866  6458  5587  5951     NA
##  8 25-34 Male   2009 20440   184   695  1806  6495  5508  5752     NA
##  9 25-34 Male   2008 20210   172   714  1874  6356  5277  5816     NA
## 10 25-34 Male   2007 20024   246   757  1930  6361  5137  5593     NA
## # ... with 356 more rows
```

```r
edu_t <- gather(data = edu,
                key = school,
                value = freq,
                elem4:coll4)

head(edu_t)
```

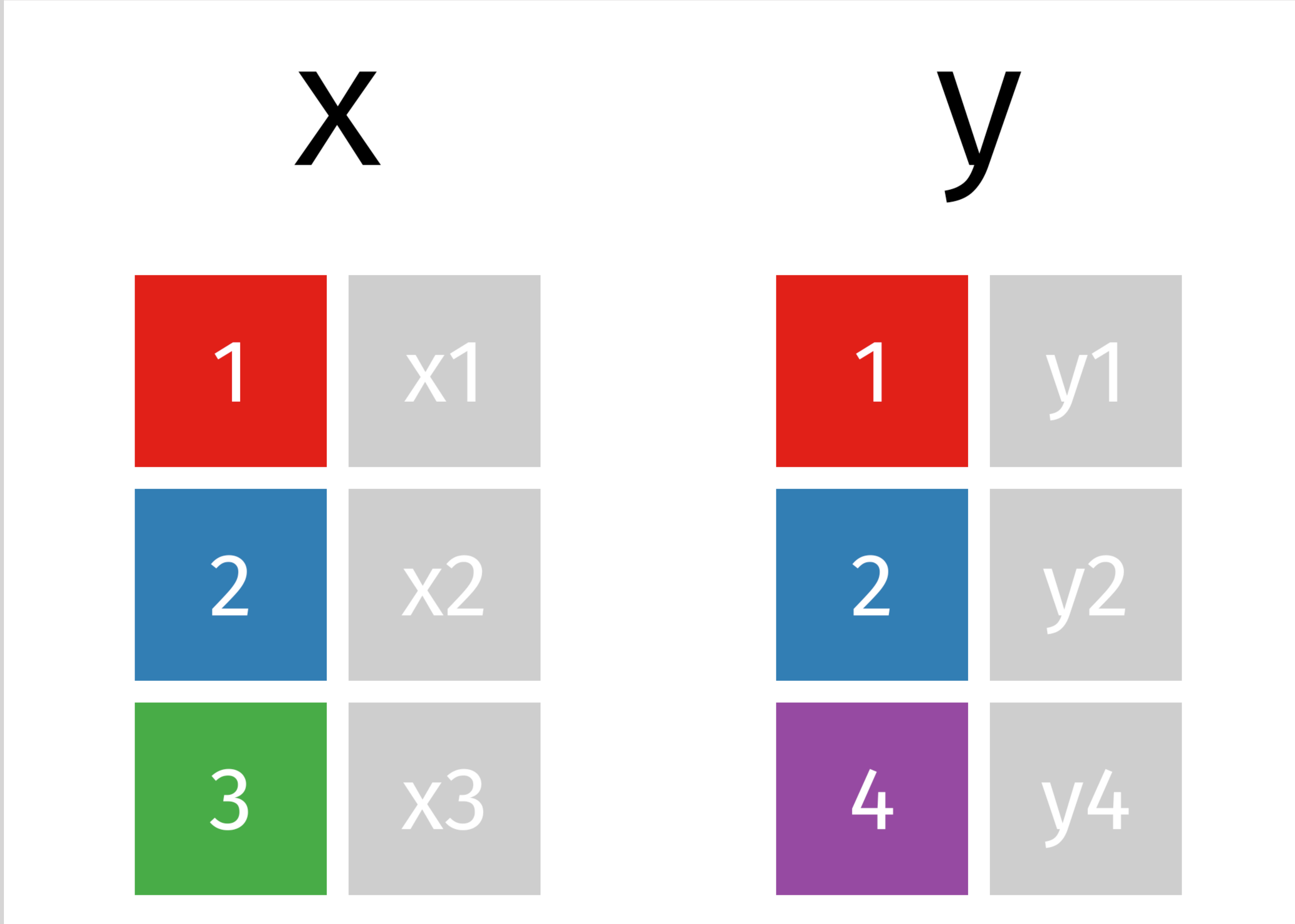
```
## # A tibble: 6 x 7
##   age   sex    year total median school  freq
##   <chr> <chr> <int> <int>  <dbl> <chr>  <dbl>
## 1 25-34 Male   2016 21845     NA elem4    116
## 2 25-34 Male   2015 21427     NA elem4    166
## 3 25-34 Male   2014 21217     NA elem4    151
## 4 25-34 Male   2013 20816     NA elem4    161
## 5 25-34 Male   2012 20464     NA elem4    161
## 6 25-34 Male   2011 20985     NA elem4    190
```

```r
tail(edu_t)
```

```
## # A tibble: 6 x 7
##   age   sex     year total median school  freq
##   <chr> <chr>  <int> <int>  <dbl> <chr>  <dbl>
## 1 55>   Female  1959 16263   8.30 coll4    688
## 2 55>   Female  1957 15581   8.20 coll4    630
## 3 55>   Female  1952 13662   7.90 coll4    628
## 4 55>   Female  1950 13150   8.40 coll4    436
## 5 55>   Female  1947 11810   7.60 coll4    343
## 6 55>   Female  1940  9777   8.30 coll4    219
```

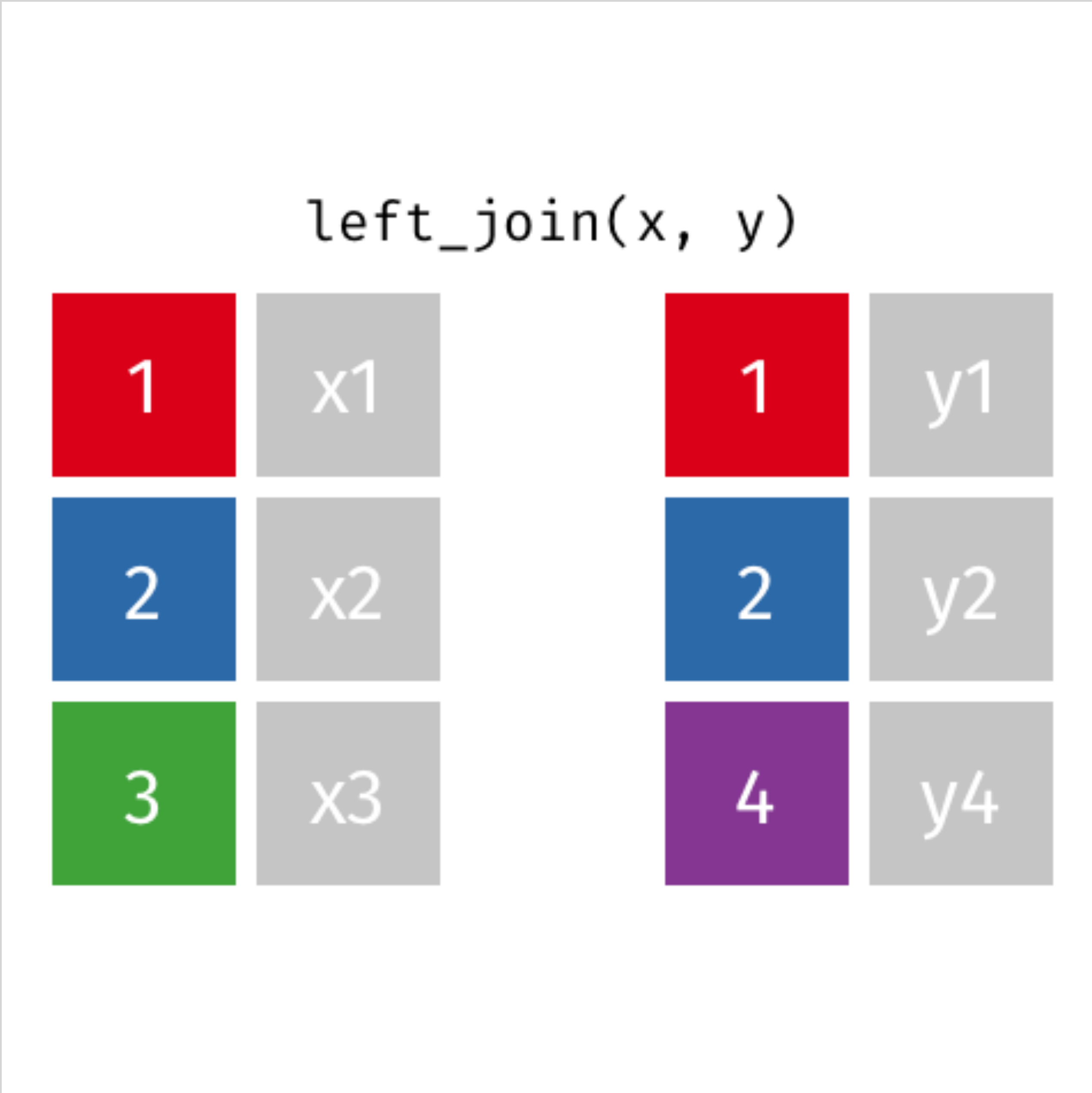
~/>_

TABLE JOINS



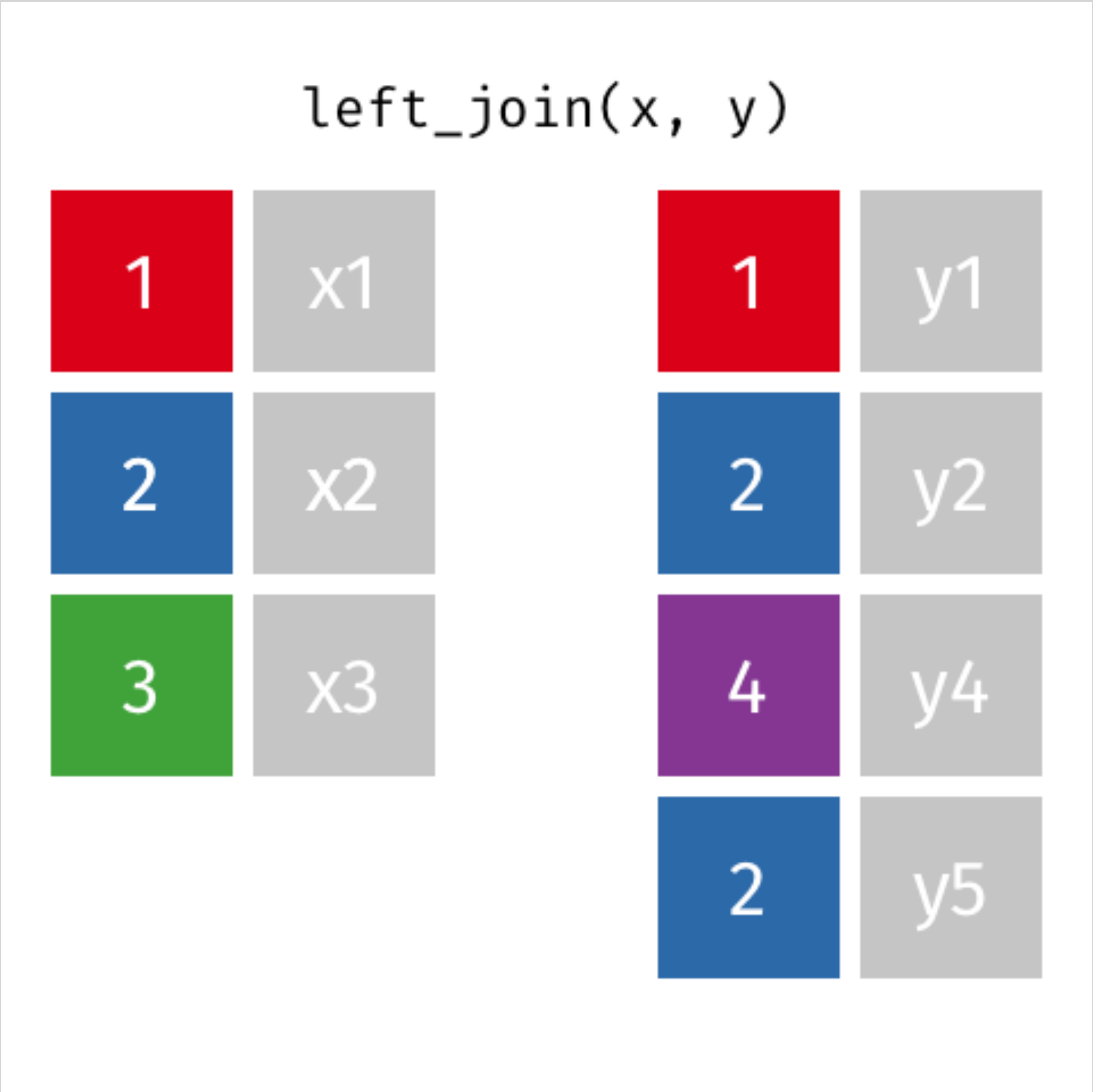
Spiffy Join Animatations courtesy Garrick Aden-Buie:

`github.com/gadenbuie/join-animations-with-gganimate.R`



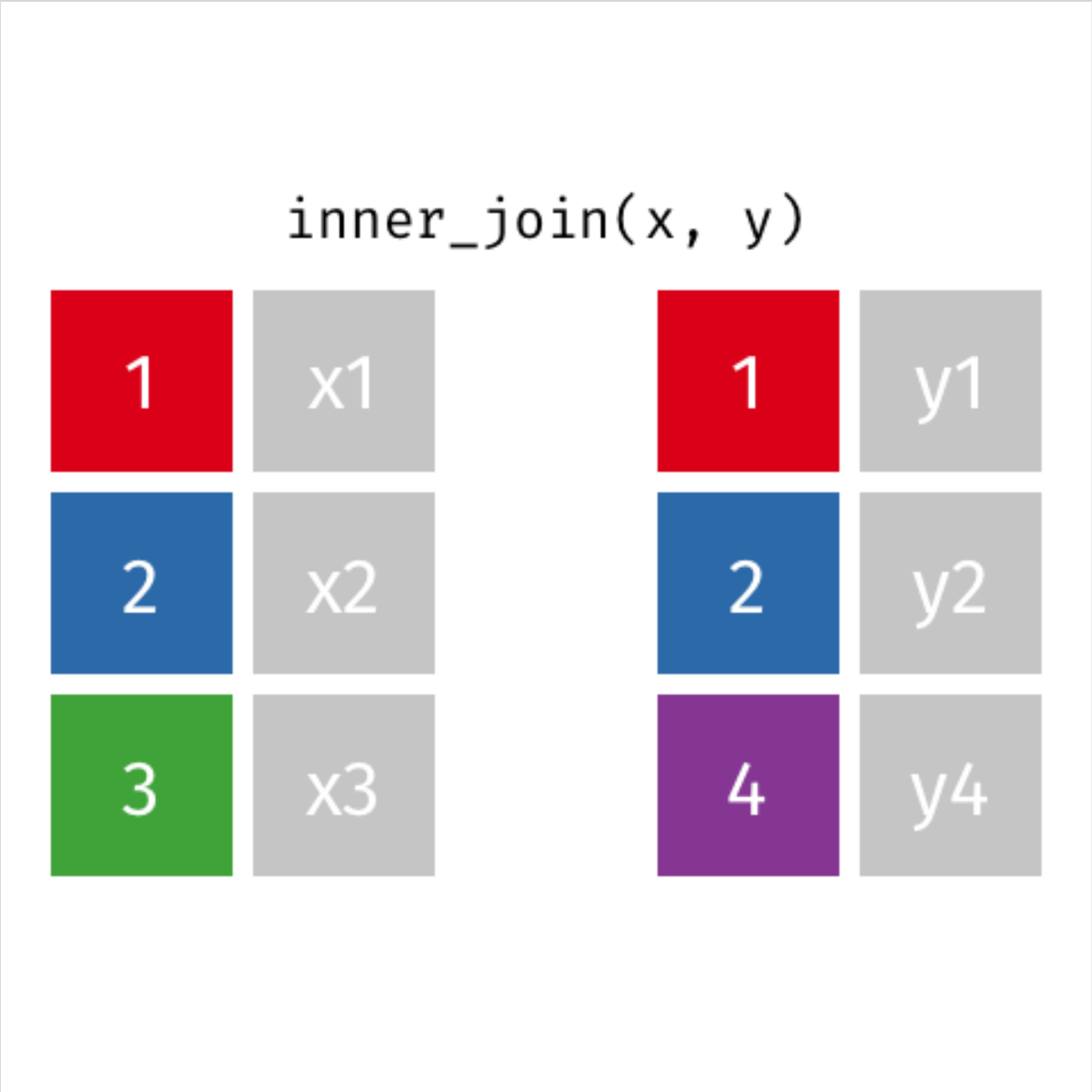
All rows from x, and all columns from x and y. Rows in x with no match in y will have NA values in the new columns.

LEFT JOIN



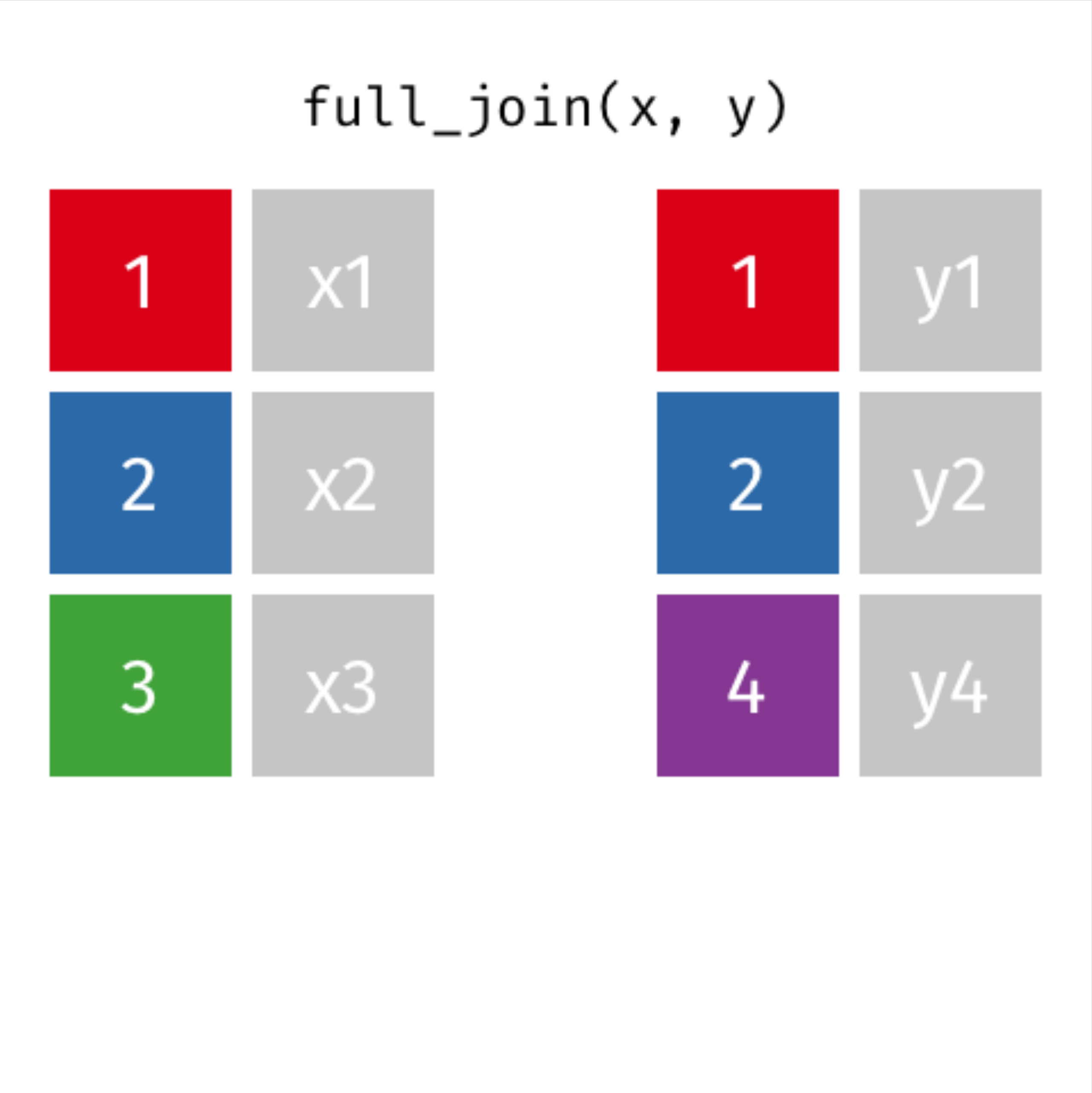
If there are multiple matches between x and y, all combinations of the matches are returned.

LEFT JOIN



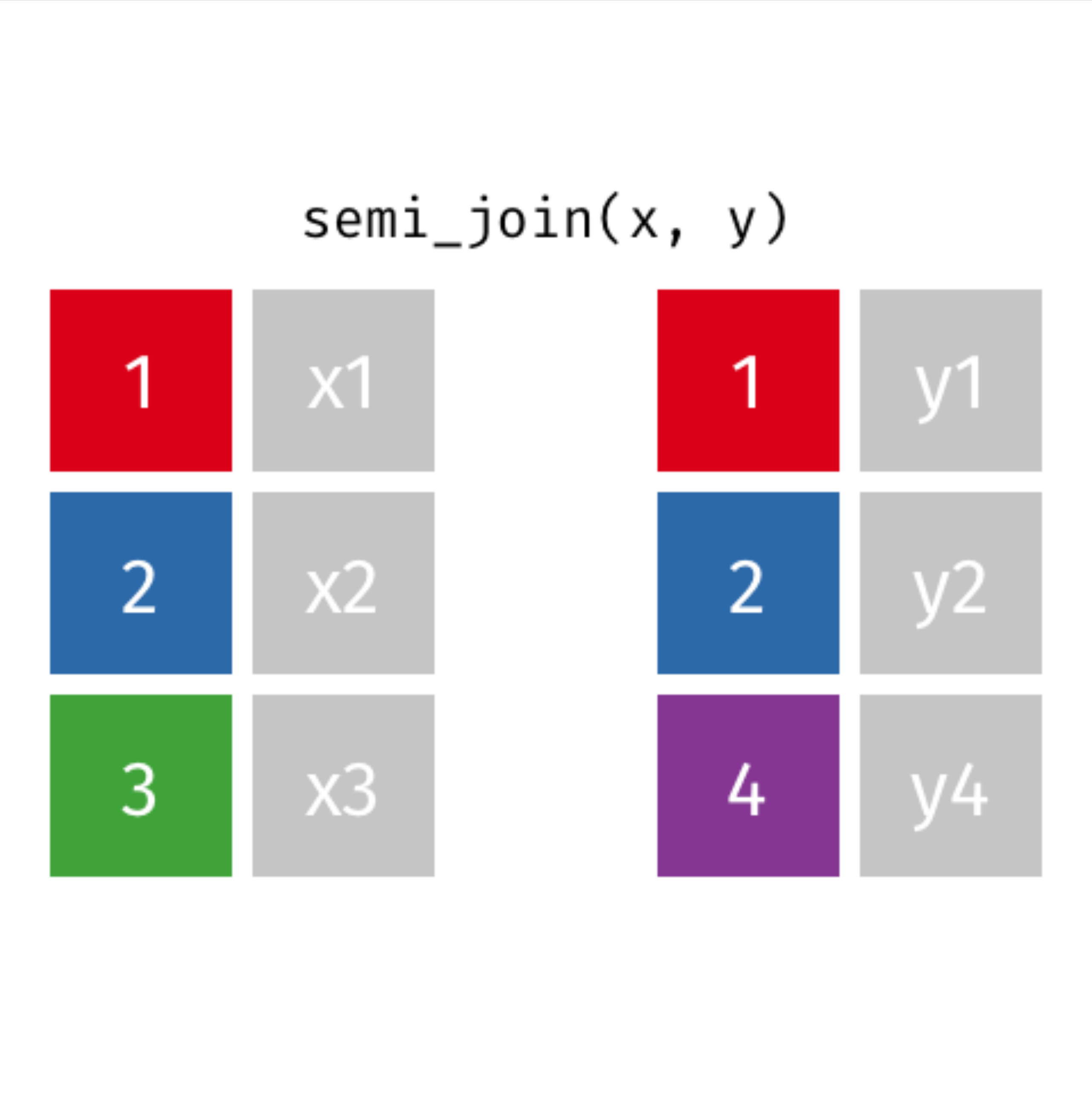
All rows from x where there are matching values in y, and all columns from x and y.

INNER JOIN



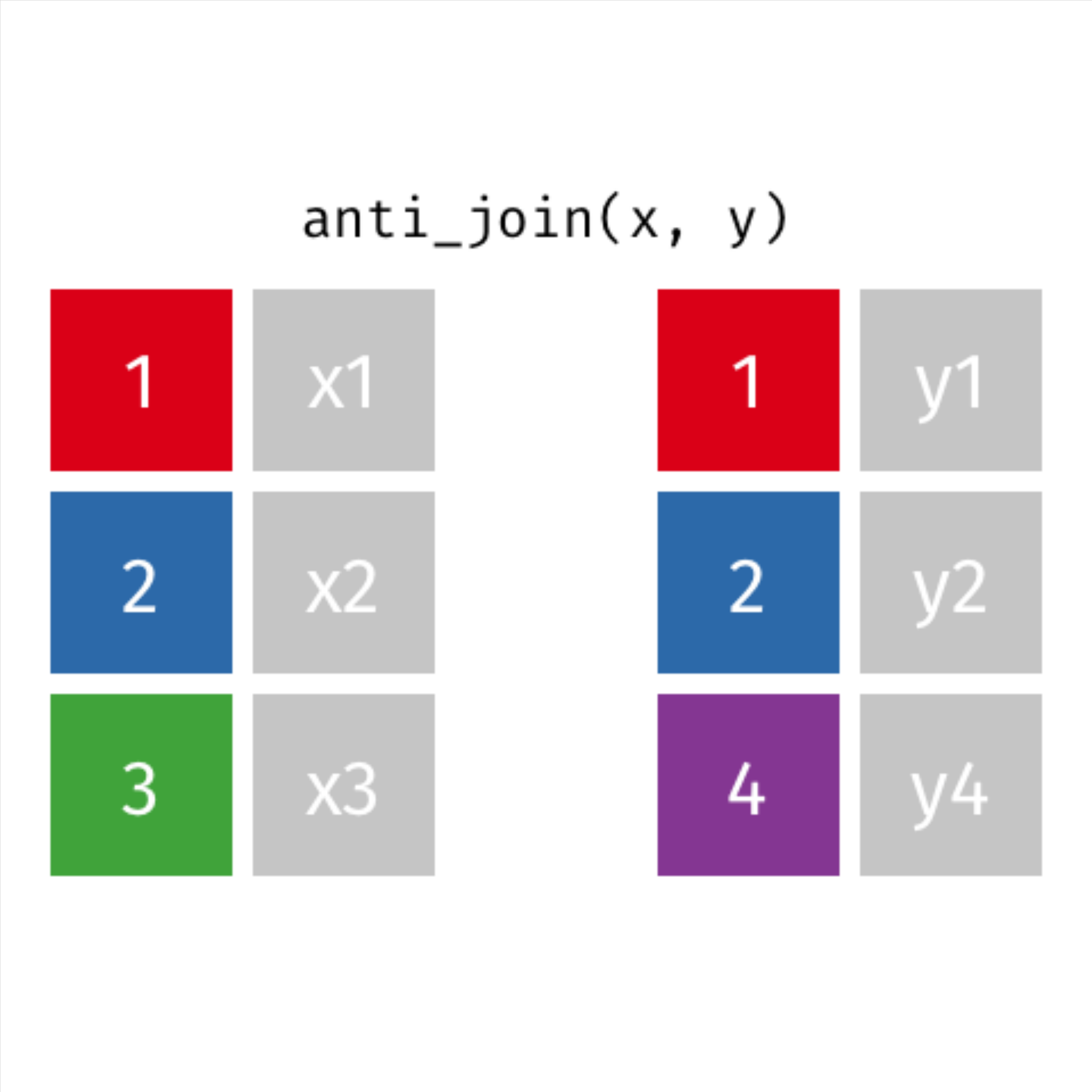
All rows and all columns from both x and y. Where there are not matching values, returns NA for the one missing.

FULL JOIN



All rows from x where there are matching values in y, keeping just columns from x.

SEMI JOIN



All rows from x where there are not matching values in y, keeping just columns from x.

ANTI JOIN

```r
senate <- data %>%
  filter(position == "U.S. Senator") %>%
  group_by(pid) %>%
  summarize(first = first(first),
            last = first(last),
            party = first(party),
            state = first(state),
            start = first(start),
            end = first(end))

house <- data %>%
  filter(position == "U.S. Representative") %>%
  group_by(pid) %>%
  summarize(state = first(state),
            district = first(district),
            start = first(start),
            end = first(end))
```

```r
sen_and_house <- inner_join(senate, house, by = "pid")
```

```
str_detect(string, pattern)

str_replace(string, pattern, replacement)
```

STRINGR

Joseph Roso 9:30 AM

I've got a tangential question: we used the following code to create the full_name variable:

Untitled

```
mutate(full_name = paste(first, last, suffix))
```

However, I noticed that when the suffix is NA, that is pasted onto the end of the name. So the full_name variable is filled with "Jon Doe NA." Is there a clever way to tell paste() to paste nothing if the suffix is NA?

kjhealy 9:34 AM

Hm.

```
str_detect(string, pattern)

str_replace(string, pattern, replacement)
```

STRINGR

```
mutate(full_name = paste(first, last, suffix),
       full_name = str_remove(full_name, " NA$"))
```

STRINGR

STRINGR