DATA MANAGEMENT IN R: AN INTRO TO THE TIDYVERSE

Carolyn Coberly

November 2017

GOALS

- Create an RMarkdown file
- Create a dataset
  - Tibbles and tribbles
- Modify a dataset
  - Modify variables and observations
  - Reshape a dataset
- Merge datasets

R MARKDOWN

- Publishes your code and comments as an html or pdf file.
- A best practice for transparent data publishing.
- Select dropdown menu File/New File/R Markdown…
  - Choose the type of file you would like to publish
- Regular text (your comments) will publish as normal and code is marked in blocks
  - Here, # indicates a section head (## a subsection, and so-forth)

WRITING CODE IN MARKDOWN

- R code must begin and end with back quotation marks: ```
- In {r name, options} write the program used (r) and the name of this block of code. Publishing options follow the comma.
  - include=FALSE will not print this segment of code or its results in the final document
  - echo=FALSE prints the results, but not the code in the final document
  - message=FALSE will not print messages generated by the code
  - warning=FALSE will not print warnings generated by the code
- Enter the code as normal within the block.

RUNNING AND PUBLISHING A FILE

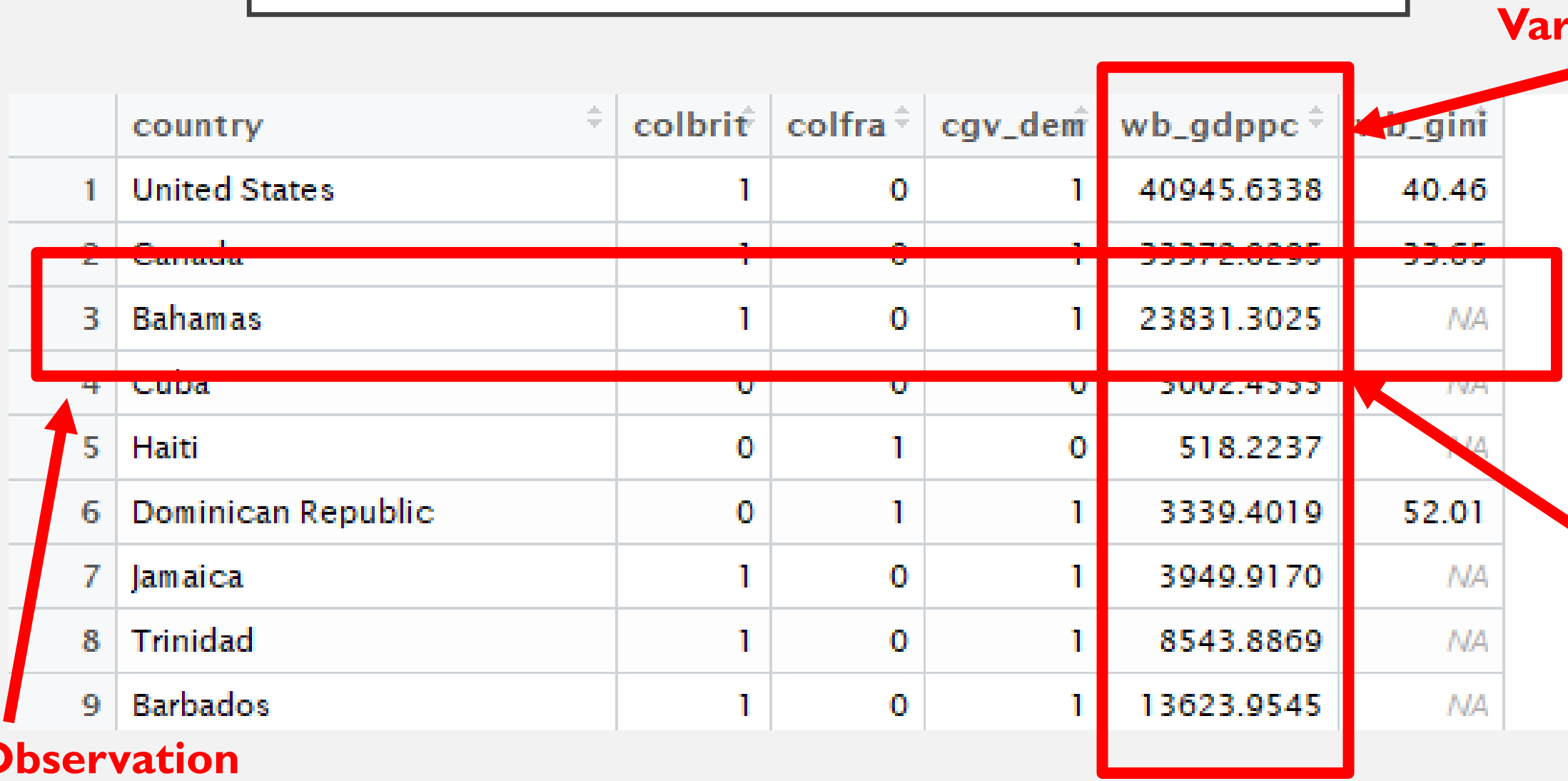
- You can run an individual code block by clicking on theright arrow in its top right corner.

  You can run all code blocks *before* the current block by clicking on the down arrow.
- Click "knit" when you are ready to publish the document.
  - This takes time, so wait until the end.

CREATE A DATASET

- Tibbles
  - Create datasets by inputting formulae for variables
- Tribbles
  - Create datasets with customized input of data
- Best used for adding individual variables or small datasets

VARIABLES, VALUES, AND OBSERVATIONS



	country	colbrit	colfra	cgv_dem	wb_gdppc	wb_gini
1	United States	1	0	1	40945.6338	40.46
2	Canada	1	0	1	33372.8295	33.65
3	Bahamas	1	0	1	23831.3025	NA
4	Cuba	0	0	0	9062.4333	NA
5	Haiti	0	1	0	518.2237	NA
6	Dominican Republic	0	1	1	3339.4019	52.01
7	Jamaica	1	0	1	3949.9170	NA
8	Trinidad	1	0	1	8543.8869	NA
9	Barbados	1	0	1	13623.9545	NA

MODIFYING VARIABLES AND OBSERVATIONS

- **Changing variable names**
  - data <- rename(data, newname = oldname)
- **Changing variable type**
  - character, numeric, boolean, factor
  - variable <- as.character(variable)
- **Modifying values**
  - To change all observations in a variable
    - variable <- recode(variable, oldvalue = newvalue)
  - To change observations based on variable characteristics
    - variable[logical command] <- newvalue

ADDING AND DELETING VARIABLES

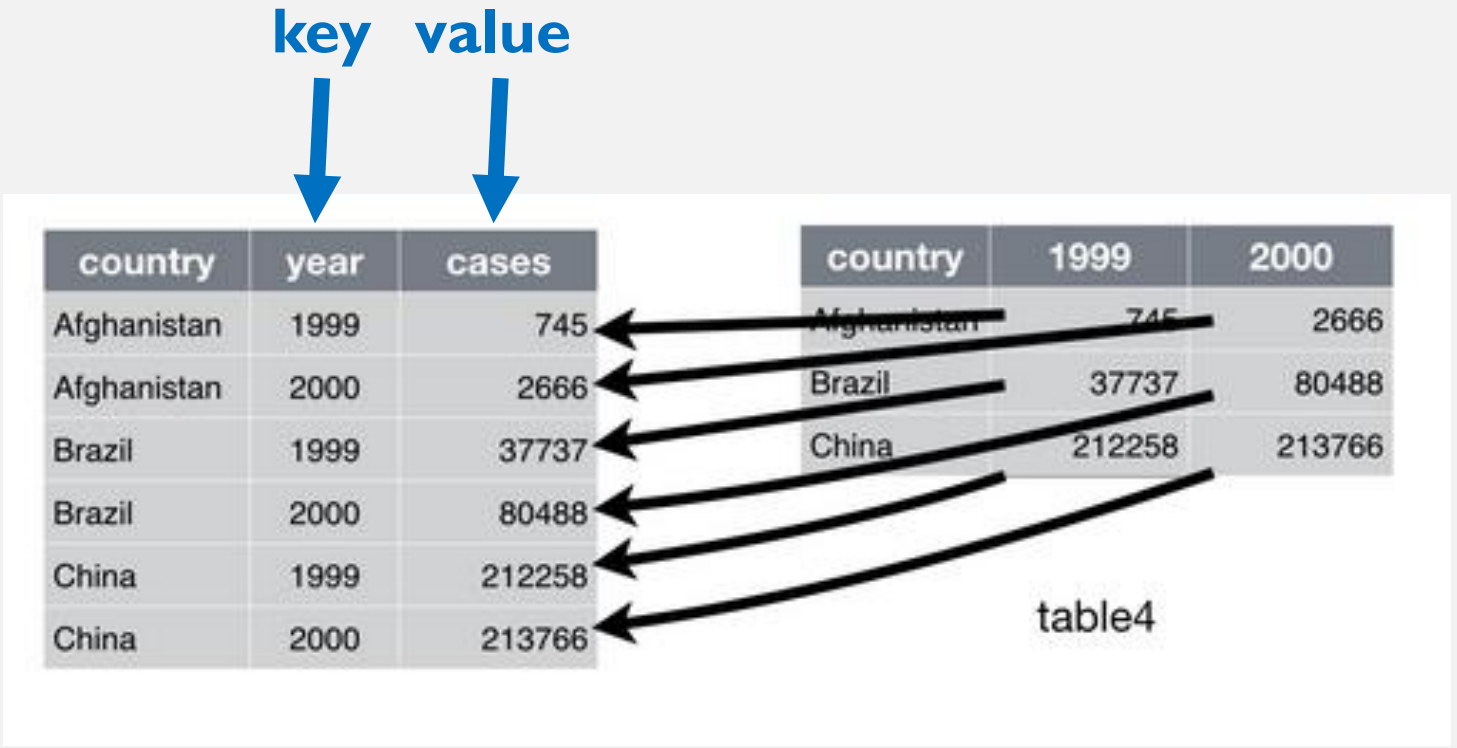
- Deleting observations: Filter
  - data <- filter(data, logical commands)
    - Data <- filter(data, year>=1950)
- Deleting variables: Select
  - Delete a single variable (-)
    - data <- select(data, -variable)
  - Select variables to keep
    - data <- select(data, keep1, keep2, keep3)
- Adding variables: Mutate
  - mutate(data, newvar = var1 / var2)
  - You can use many types of equations
  - You do not need to assign this function to an object

CHANGING THE SHAPE OF A DATASET

- Sometimes data is prepared so that data you want as observations are presented as variables (and vice versa).
- In order to reshape the dataset so that it presents the data the way you need, use the function gather or spread.
- The key concepts to understand in the code are keys and values.

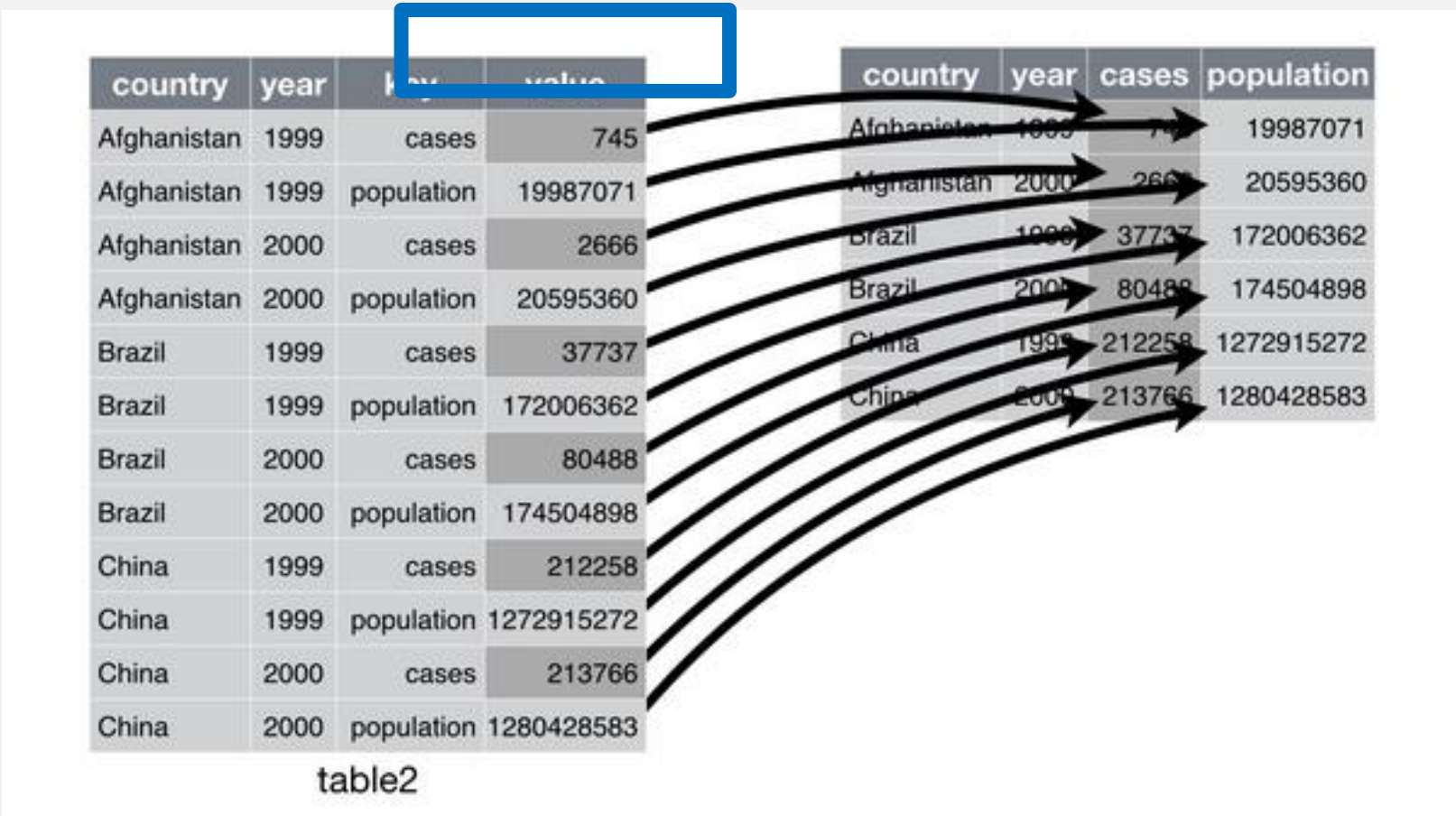
GATHER

- Gather converts variables into observations (changing a wide dataset into a long one).



SPREAD

- Spread converts observations into variable



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

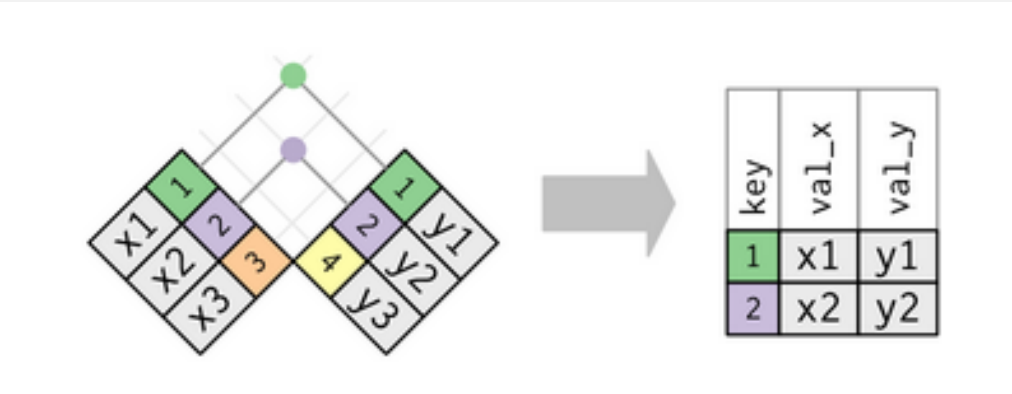
table2

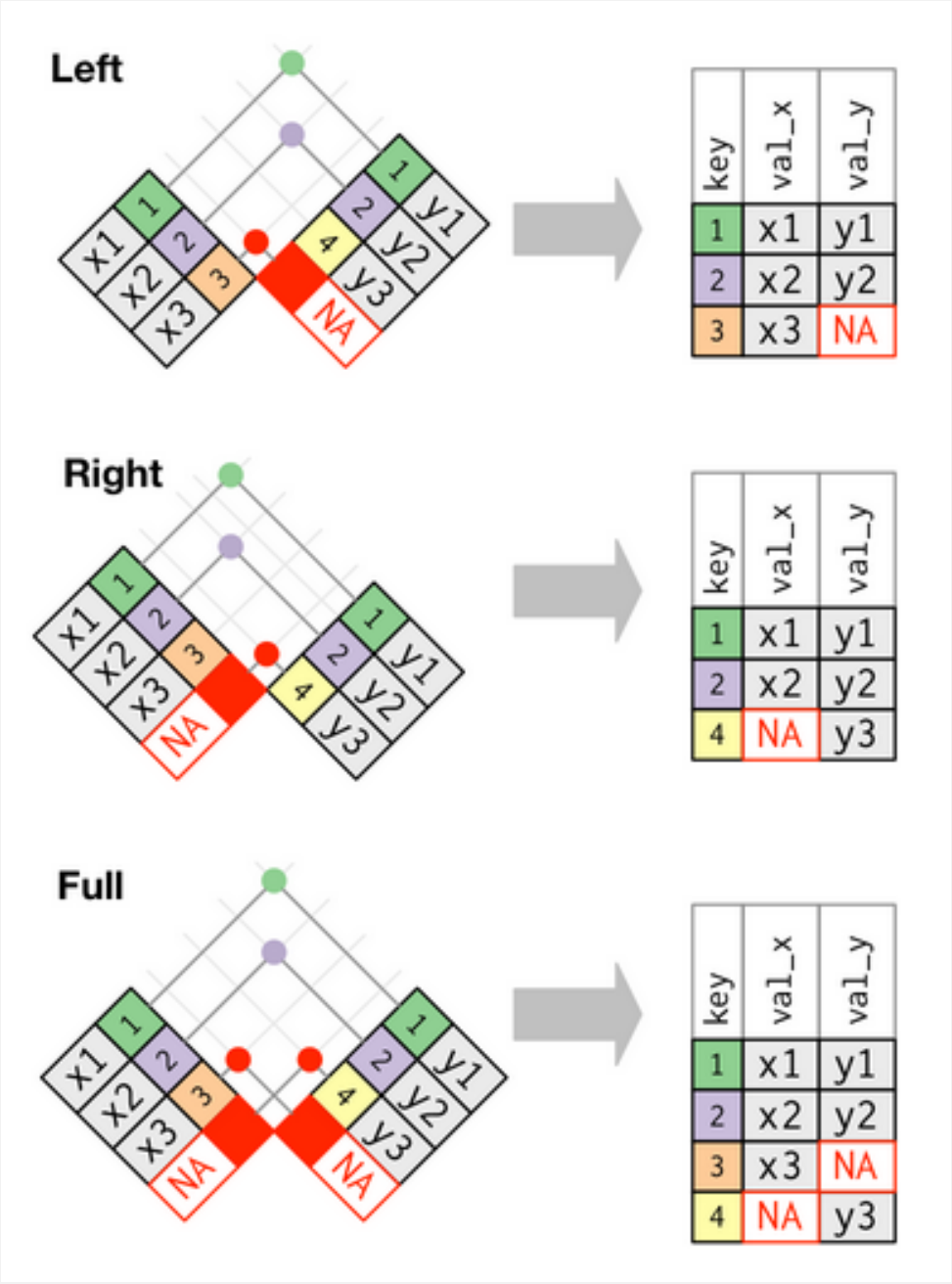
country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

MERGING DATASETS

- There are four types of merges you can do using tidyr functions:
  - An *inner* join keeps only the observations that both datasets share
  - *Left* and *right* joins keep all the observations in the first (left) or second (right) datasets and adds observations from the other dataset.
  - A *full* join keeps all observations from both datasets.
- In the code, specify the two datasets you want to merge and the variable (key) you want to use to do it.
  - Merge <- full_join(data1, data2)
  - Merge <- full_join(data1, data2, by = "key")
  - Merge <- full_join(d1, d2, by = c("key1" = "key2", "key3" = "key4"))

Inner Join


x1
x2
x3
y1
y2
y3
1
2
3
4
key
val_x
val_y
1 x1 y1
2 x2 y2


Left
x1
x2
x3
NA
y1
y2
y3
key
val_x
val_y
1 x1 y1
2 x2 y2
3 x3 NA
Right
x1
x2
x3
NA
y1
y2
y3
key
val_x
val_y
1 x1 y1
2 x2 y2
4 NA y3
Full
x1
x2
x3
NA
NA
y1
y2
y3
key
val_x
val_y
1 x1 y1
2 x2 y2
3 x3 NA
4 NA y3

COMMON PROBLEMS WITH MERGES

- Problem: Not all observations joined correctly
- Diagnosis:
  - Number of observations increases during the merge
  - Missing data from one dataset in initial observations
- Potential solutions:
  - Use a different key (country code not country name, for example)
  - Recode original dataset(s) so that keys match
  - Make sure variables are same type (character/numeric)

HELP!

- RMarkdown Tutorial (rmarkdown.rstudio.com)
- *R for Data Science* by Hadley Wickham & Garrett Grolemund (http://r4ds.had.co.nz/ and for purchase on Amazon)
- Cheat sheet: https://www.rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf