A tidy data playbook

*These slides by Garrett Grolemund based on correspondence with Hadley Wickham

Tidy data

Tidy data is a format for laying out tables of data. It is the most efficient format to use when manipulating data with R. It aligns with R's data structures and works well with R's vectored operations.

These slides outline a strategy for tidying untidy data.

Definitions

Definitions

Variable
A quantity, quality, or property that you can measure.

Value
The state of a variable that you observe when you measure it.

Observational Unit
The type of object that you measure when you measure a variable.

Observation
A set of values that display the relationship between variables (this relationship could be no relationship). To be an observation, values need to be measured under similar conditions, usually measured on the same observational unit at the same time.

Definitions

Data
A set of values, each associated with a variable and an observation.

* Since values are tied to variables and observations, the rules of tidy data are interdependent.

** This definition doesn’t play well with "unstructured data" (e.g. a batch of emails to data mine), (but I don’t think of unstructured data as data anyways. To me, it is "pre-data" at most).

The rules

Tidy data will satisfy four **interdependent** rules.
Any three of these rules imply the third…

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

1. Each variable is in its own column

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

2. Each observation is in its own row

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

3. Each value is in its own cell

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

4. Each observational unit is in its own table

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

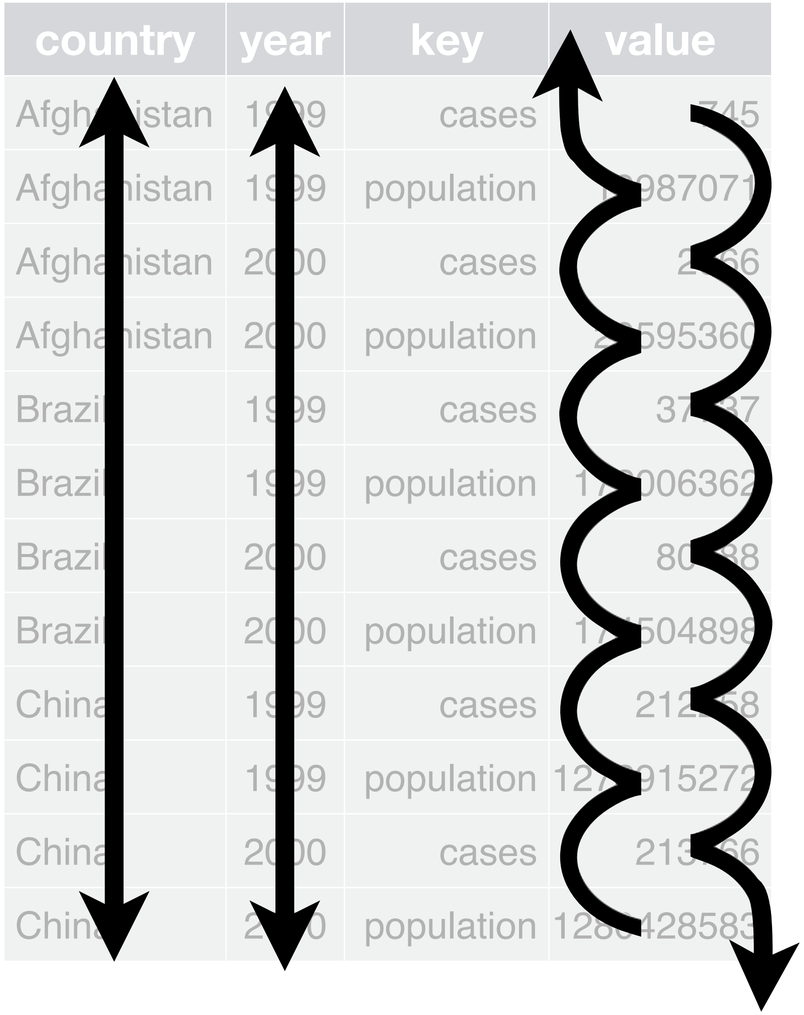
The cases of untidy data

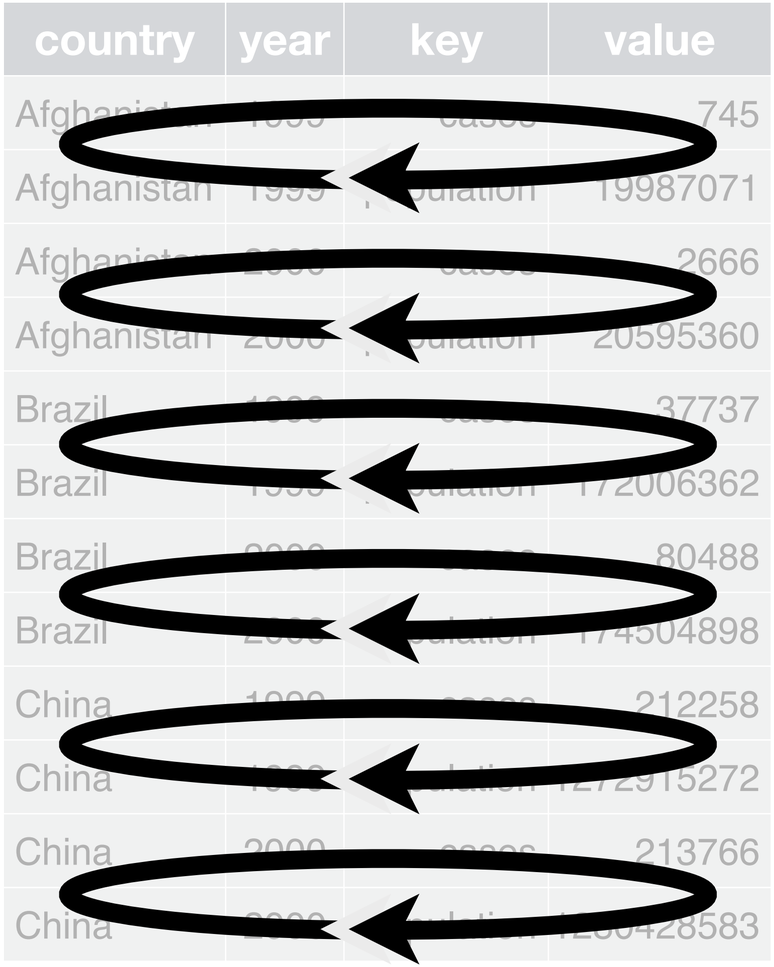
1 Too long data

Violates:

1. Each variable is in its own column (n->1)
2. Each observation is in its own row (1->n)

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





library(tidyr)

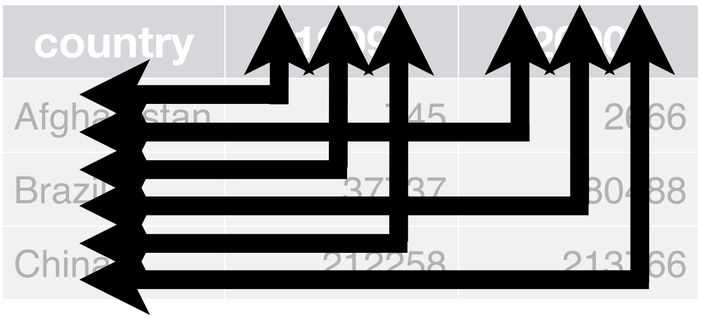
spread()

2 Too wide data "rectangular data"

Violates:

1. Each variable is in its own column (1 -> n)
2. Each observation is in its own row (n -> 1)

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



2 Too wide data "rectangular data"

library(tidyr)

gather()

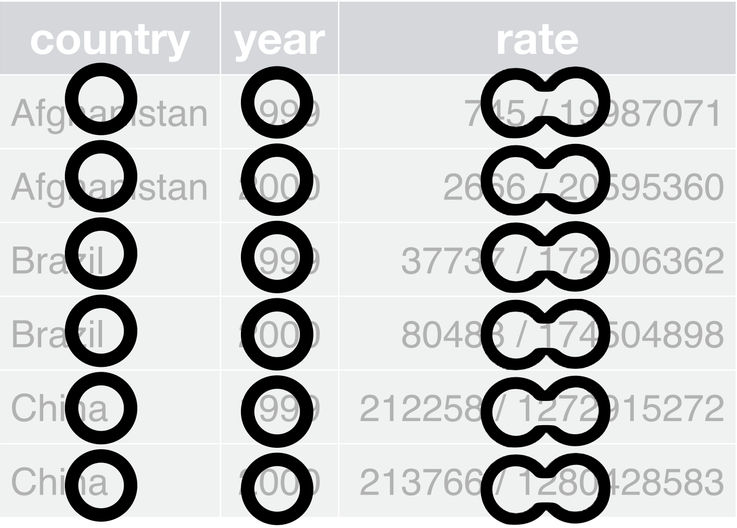
3 Combined values

Violates:

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

1. Each variable is in its own column (n->1)

3. Each value is in its own cell (n->1)



library(tidyr)

separate()

4 Split values

Violates:

storm	year	month	day
Alberto	2000	08	12
Alex	1998	07	30
Allison	1995	06	04
Ana	1997	07	1
Arlene	1999	06	13
Arthur	1996	06	21

1. Each variable is in its own column (1->n)

storm	year	month	day
Alberto	2000	08	12
Alex	1998	07	30
Allison	1995	06	04
Ana	1997	07	1
Arlene	1999	06	13
Arthur	1996	06	21

3. Each value is in its own cell (1->n)

storm	year	month	day
Alberto	2000	08	12
Alex	1998	07	30
Allison	1995	06	04
Ana	1997	07	1
Arlene	1999	06	13
Arthur	1996	06	21

library(tidyr)

unite()

5 Combined types

country	male	female	0-20	21-40	41-60
NZ	0.18	0.86	0.87	0.48	0.79
USA	0.17	0.40	0.89	0.02	0.11

Violates:

4. Each observational unit is in its own table (n-> 1)

female	male	country	0-20	21-40	41-60
0.86	0.18	NZ	0.87	0.48	0.79
0.40	0.17	USA	0.89	0.02	0.11

2. Each observation is in its own row (n->1)

female	male	country	0-20	21-40	41-60
0.86	0.18	NZ	0.87	0.48	0.79
0.40	0.17	USA	0.89	0.02	0.11

library(dplyr)

select(), filter()

6 Split types

country	year	cases
Afghanistan	2000	2666
Brazil	2000	80488
China	2000	213766

country	year	population
Afghanistan	2000	20595360
Brazil	2000	174504898
China	2000	1280428583

Violates:

4. Each observational unit is in its own table (1->n)

country	year	cases
Afghanistan	2000	2666
Brazil	2000	80488
China	2000	213766

country	year	population
Afghanistan	2000	20595360
Brazil	2000	174504898
China	2000	1280428583

2. Each observation is in its own row (1->n)

country	year	cases
Afghanistan	2000	2666
Brazil	2000	80488
China	2000	213766

country	year	population
Afghanistan	2000	20595360
Brazil	2000	174504898
China	2000	1280428583

library(dplyr)

full_join()

Very messy data

Basic order

If your data is untidy in multiple ways, tidy it in this order…

The first steps prevent R from creating meaningless values with vectorized operations. They ensure that each row contains values from the same observation.

1. **Make sure that each data table contains only one type of observational unit**
   i. Unjoin types combined in the same table
   ii. Join types split across several tables

2. **Gather together untidy columns**
   i. If you need to work on the variable names, temporarily treat them as values: gather them into their own column

The next steps make it possible for R to easily access values and variables.

3. **Make sure each cell only contains one value**
   i. Separate values combined in the same cell
   ii. Unite values split across several cells

4. **Finally, spread each variable into its own column**