# Atidy data playbook

### 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).

## It he 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
Afghanstan	1919	745	19937071
Afghanistan	2000	2566	20595360
Brazil	1999	37737	1720)6362
Brazil	2000	80488	1745)4898
China	1999	212258	12729   5272
China	200	213 66	12804 28583

### 2. Each observation is in its own row



## 3. Each value is in its own cell

country	year	cases	population
Afglanstan	199	745	19987071
Afglanstan	200	2666	2069:360
Braul	199	3 737	172006362
Brau	200	8/488	174604898
Chira	199	21(238	1272613272
Chira	200	21776	1280 728583

### 4. Each observational unit is in its own table

year	cases	population
1999	745	19987071
2000	2666	20595360
1999	37737	172006362
2000	80488	174504898
1999	212258	1272915272
2000	213766	1280428583
	1999 2000 1999 2000 1999	<ul> <li>2000 2666</li> <li>1999 37737</li> <li>2000 80488</li> <li>1999 212258</li> </ul>

# The cases of untidy data

country

Afghanistan 1999

Afghanistan 1999

Afghanistan 2000

Afghanistan 2000

Brazil

Brazil

Brazil

Brazil

China

China

China

China

#### Too long data

year

1999

1999

2000

2000

1999

2000

2000

key

population

population

population

population

cases

cases

cases

cases

cases

cases

1999 population 1272915272

population 1280428583

value

19987071

20595360

172006362

174504898

745

2666

37737

80488

212258

213766

## Violates:

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

country	year	key	<b>Value</b>
Afghaistan	1209	cases	745
Afghanistan	1999	population	987071
Afghanistan	2000	cases	2.66
Afghanistan	2000	population	595360
Brazil	1999	cases	37 37
Brazil	1999	population	1>006362
Brazil	2000	cases	80.88
Brazil	2000	population	17504898
China	1999	cases	212.58
China	1999	population	127 915272
China	2000	cases	213.56
China	20	population	1285428583

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

country	year	key	value
Afghaniat	1000	Casco	745
Afghanistan	1999	Dulation	19987071
Afghaniata	2000		2666
Afghanistan	2000	Dulation	2666
Brazil	1000		37737
Brazil	1335	Julation	37737
Brazil	0000		80488
Brazil	2003	<b>Salation</b>	174504898
China	1000		212258
China	1003		1212915272
China	2000	00000	213766
China	2002	(audati	1200428583

library(tidyr)

## spread()

1999

745

37737

212258

2000

2666

80488

213766

country

Afghanistan

Brazil

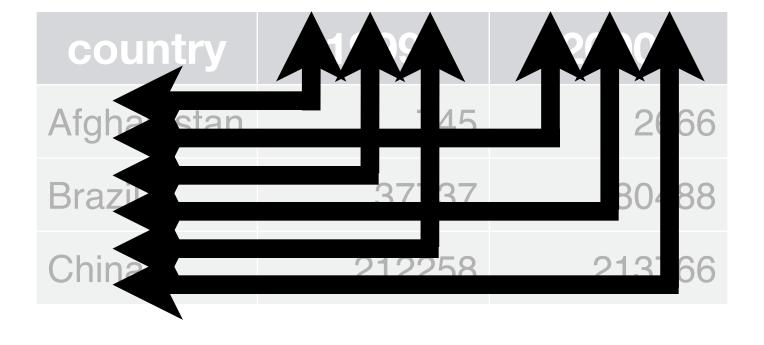
China

## Violates:

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

country		200
Afghaistan	7 10	2666
Brazi	37737	804 8
Chin	212236	213766

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



# 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)

country	year	rate
Afghanstan	1300	745 / 1986 7071
Afghanistan	2000	2666 / 2059 5360
Brazil	1999	37'37 / 1720(6362
Brazil	2000	80-88 / 1745(4898
China	1999	212258 / 12729 5272
China	200	213 6 / 12804 3583

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

country		rate
Afghanstan	999	745 / 19987071
Afgranstan	100	2666 / 20595360
Bravil	99)	37737 / 172)06362
Bravil		80488 / 174504898
China	99)	212258/1272915272
China	(00)	213766/1280428583

library(tidyr)

## separate()

## 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	00	12
Alex	1800	07	00
Allison	1995	00	04
Ana	1897	07	
Arlene	1999	00	13
Arthur	1830	00	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()

## Combined types

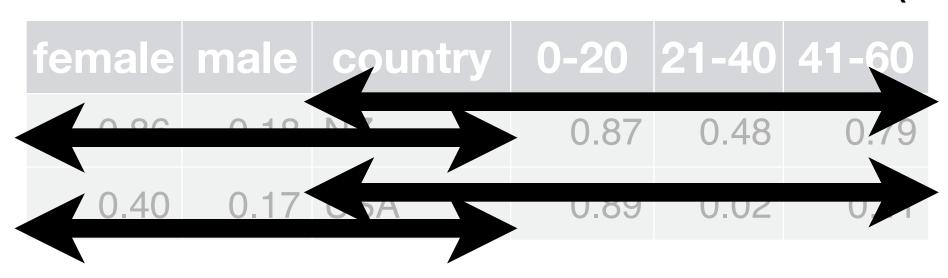
## country male female 0-20 21-40 41-60 NZ 0.18 0.86 0.87 0.48 0.79 USA 0.17 0.40 0.89 0.02 0.11

## 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	0.1	USA	0.89	0.02	0.11

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



library(dplyr)

## select(), filter()

## Split types

## 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

country	year	cases	country	year	population
Afghanistan	2000	2666	Afghanistan	2000	20595360
Brazil	2000	80488	Brazil	2000	174504898
China	2000	213766	China	2000	1280428583

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

country	year	cases	country	year	population
ngnamstan	2000	2000	Aignamstan	2000	203333
Irazii	2000	00400	DIGZII	2000	174304623
nina	2000	213/66	Unina	2000	1280428533

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