Tidy Data with R

- *Data Science with R* by Garrett Grolemund
- Chapter: Data Tidying

DSR Example Data Sets

- The data sets in the DSR Package show the same data, but organized in four entirely different ways.
- Each data set shows the same values of four variables country, year, population, and cases, but each data set organizes the values into a different layout.

Tidy Data with R

- To install DSR, run the command

```
install.packages(c("tidyr", "devtools"))

devtools::install_github("garrettgman/DSR")
```

Tidy Data with R- Data Set 1

```
library(DSR)
# Data set one
table1
## Source: local data frame [6 x 4]
##
##       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
```

Tidy Data with R

- In `table1` (previous slide), each variable is placed in its own column, each observation in its own row, and each value in its own cell.
- It properly complies with the *Principles of Tidy Data*.

- On the next set of slides - we will look at three more layouts (with the last layout comprising two separate tables)

Table 2 (sed)

```
# Data set two
table2

## Source: local data frame [12 x 4]
##
##         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
..........
```

Tidy Data with R

```
# Data set three
table3

## Source: local data frame [6 x 3]
##
##       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
```

Tidy Data with R

The last data set is a collection of two tables: cases and populations

```
# Data set four
table4  # cases

## Source: local data frame [3 x 3]
##
##       country   1999   2000
## 1 Afghanistan    745   2666
## 2      Brazil  37737  80488
## 3       China 212258 213766
```

Tidy Data with R

```
table5  # population

## Source: local data frame [3 x 3]
##
##       country        1999        2000
## 1 Afghanistan    19987071    20595360
## 2      Brazil   172006362   174504898
## 3       China  1272915272  1280428583
```

Using this data for analysis

- Assume that in these data sets, cases refers to the number of people diagnosed with TB per country per year.
- To calculate the rate of TB cases per country per year (i.e, the number of people per 10,000 diagnosed with TB), you will need to do four separate approach with the data, one for each layout.

Tidy Data with R

Each approach will do the following

1. Extract the number of TB cases per country per year
2. Extract the population per country per year (in the same order as above)
3. Divide cases by population
4. Multiply by 10000

Data set one

Since *table1* is organized in a tidy fashion, you can calculate the rate like this,

```
# Data set one
table1$cases / table1$population * 10000
```

Quick and Relatively Simple

Data set two

- Data set two intermingles the values of population and cases in the same columns.
- As a result, you will need to untangle the values whenever you want to work with each variable separately.
- Youll need to perform an extra step to calculate the rate.

Tidy Data with R

```
# Data set two
case_rows <- c(1, 3, 5, 7, 9, 11, 13, 15, 17
pop_rows <- c(2, 4, 6, 8, 10, 12, 14, 16, 18

table2$value[case_rows]
   / table2$value[pop_rows] * 10000
```

Not overly complicated, but requires specification of rows (may or may not be automatable)

Data set three

- Data set three combines the values of cases and population into the same cells.
- It may seem that this would help you calculate the rate, but that is not so.
- You will need to separate the population values from the cases values if you wish to conduct an analysis them.
- This can be done, but not with basic R syntax.

Data set four

- Data set four stores each variable in a different format: as a column, a set of column names, or a field of cells.
- As a result, you will need to work with each variable differently.

Data set four

- This makes code written for data set four hard to generalize.
- The code that extracts the values of year, `names(table4)[-1]`, cannot be generalized to extract the values of population, `c(table5$1999, table5$2000, table5$2001)`.

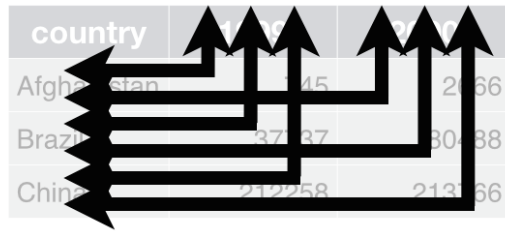
country	1999	2000
Afghanistan	745	2666
Brazil	37737	80488
China	212258	213766

table4

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

table5

variables



observations

Tidy Data with R

- The organization of data set four is inefficient in a second way as well.
- Data set four separates the values of some variables across two separate tables.
- This is inconvenient because you will need to extract information from two different places whenever you want to work with the data.