

Tidy data

PS239T

Spring 2019

- I adapted the following content from Wickham's R data science book (2017) chapter: <https://r4ds.had.co.nz/tidy-data.html> and his earlier paper (2014) published in Journal of Statistical Software: <http://www.jstatsoft.org/v59/i10/paper>.
- Wickham created so many essential packages used in R for data science. For more information, please see the following website: <http://hadley.nz/>

1. What's tidy principles

1. The dataset: strucure (physical layout) + semantics (meaning)

1.1. Data structure

- rows and columns

1.2. Data semantics

- variables > values (numbers or strings)

2. Tidy tada: “provide a standard way to organize data values within a dataset”

2.1 Tidy principles

- 1. Each variable forms a column.
- 2. Each observation forms a row.
- 3. Each type of observational unit forms a table.

```
# load library
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.2.1 --
```

```
## v ggplot2 3.1.0      v purrr  0.2.5
## v tibble  1.4.2      v dplyr  0.7.8
## v tidyr   0.8.2      v stringr 1.3.1
## v readr   1.2.1      v forcats 0.3.0
```

```
## -- Conflicts ----- tidyverse_conflicts() --
```

```
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
# tidy data example
table1
```

```
## # A tibble: 6 x 4
##   country      year  cases population
##   <chr>      <int> <int>      <int>
## 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
```

Practically, this approach is good because you're going to have consistency in the format of data across all the projects you're working on. Also, tidy data works well with key packages (e.g., dplyr, ggplot2) in R.

Computationally, this approach is good for vectorized programming because "different variables from the same observation are always paired".

3. Messy datasets

3.1. Signs of messy datasets

- 1. Column headers are values, not variable names.
- 2. Multiple variables are not stored in one column.
- 3. Variables are stored in both rows and columns.
- 4. Multiple types of observational units are stored in the same table.
- 5. A single observational unit is stored in multiple tables.

4. Tidy tools

Either input or output or both of them can be messy. Tidy tools can fix these problems.

4.1. Manipulation

- Filter
- Transform
- Aggregate
- Sort

4.1.1. Gather (from wide to long)

```
table4a
```

```
## # A tibble: 3 x 3
##   country    `1999` `2000`
## * <chr>      <int> <int>
## 1 Afghanistan    745   2666
## 2 Brazil        37737  80488
## 3 China         212258 213766
```

```
# from long to wide
```

```
table4a %>%
  gather('1999', '2000', key = "year", value = "population")
```

```
## # A tibble: 6 x 3
##   country    year population
##   <chr>      <chr>      <int>
## 1 Afghanistan 1999         745
## 2 Brazil      1999        37737
## 3 China        1999       212258
## 4 Afghanistan 2000         2666
## 5 Brazil      2000       80488
## 6 China        2000      213766

# save the file
table4a_wide <- table4a %>%
  gather('1999', '2000', key = "year", value = "population")
```

4.1.2. Spread (from wide to long)

```
table4a_wide %>%
  spread(key = "year", value = "population")
```

```
## # A tibble: 3 x 3
##   country    `1999` `2000`
##   <chr>      <int> <int>
## 1 Afghanistan    745    2666
## 2 Brazil        37737  80488
## 3 China        212258 213766
```

4.1.3. Separate (split one into many columns)

```
table3

## # A tibble: 6 x 3
##   country    year rate
##   * <chr>      <int> <chr>
## 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

table3 %>%
  separate(rate, into = c("cases", "population"))
```

```
## # A tibble: 6 x 4
##   country    year cases population
##   * <chr>      <int> <chr> <chr>
## 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
```

4.1.4. Unite

4.2. Visualization

Will be covered in the future session.

4.3. Modeling

Will not be extensively covered in this course.