Tidy Data

Brad Stieber

2018-10-29

Introduction

Idea (the theory)

Execution (the practice)

Displaying and Organizing Data

Conclusion

GitHub

You can find the repository for this presentation on my GitHub.

The only package needed:

```
library(tidyverse)
```

Introduction

Goals for this talk

My goal is for you to walk away with an understanding of:

- Tidy data philosophy

Goals for this talk

My goal is for you to walk away with an understanding of:

- Tidy data philosophy
- Some `tidyverse` data terminology

Goals for this talk

My goal is for you to walk away with an understanding of:

- Tidy data philosophy
- Some `tidyverse` data terminology
- Common types of *untidy* data

Goals for this talk

My goal is for you to walk away with an understanding of:

- Tidy data philosophy
- Some `tidyverse` data terminology
- Common types of *untidy* data
- Displaying and organizing tidy data

Where did this come from?

Most of what follows is based off of Hadley Wickham's paper on tidy data.

If you're looking for a practical introduction, Hadley Wickham has one of those too.

I also borrow from other resources (listed at the end), as well as my own experience working with tidy *and* untidy datasets.

But mostly. . .

Idea (the theory)

Why tidy data?

- Consistency

Why tidy data?

- Consistency
- Rely on vectorization (in common data tools like `R` and `pandas`), and expected/desired behavior in grouped aggregation (excel, tableau)

Why tidy data?

- Consistency
- Rely on vectorization (in common data tools like `R` and `pandas`), and expected/desired behavior in grouped aggregation (excel, tableau)
- Foresight

Why tidy data?

- Consistency
- Rely on vectorization (in common data tools like `R` and `pandas`), and expected/desired behavior in grouped aggregation (excel, tableau)
- Foresight
- It's never been easier

What is tidy data?

There are three qualities a dataset must have to be considered "tidy":

1. Each variable forms a column.

What is tidy data?

There are three qualities a dataset must have to be considered "tidy":

1. Each variable forms a column.
2. Each observation forms a row.

What is tidy data?

There are three qualities a dataset must have to be considered "tidy":

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

The Language of Tidy Data

- Dataset: a collection of values (e.g. iris data)
- Variable: all values that measure the same underlying attribute (e.g. height, width)
- Values: a specific measurement or attribute for a variable (e.g. $100)
- Observation: all values measured on the same unit (like a person, or a day, or a game) across variables

It's usually easy to figure out things like *observations* and *variables* for a given dataset, but defining them in the abstract can be difficult.

The Data Tidying Operations

Getting data into a tidy format requires understanding the three qualities of tidy data, as well as the five most common types of untidy data (more on these shortly).

Then, we can get most forms of untidy data to be tidy by utilizing four verbs of data tidying

- `gather`: takes multiple columns, and gathers them into key-value pairs - it makes "wide" data longer (like `UNPIVOT` in `SQL`)

The Data Tidying Operations

Getting data into a tidy format requires understanding the three qualities of tidy data, as well as the five most common types of untidy data (more on these shortly).

Then, we can get most forms of untidy data to be tidy by utilizing four verbs of data tidying

- `gather`: takes multiple columns, and gathers them into key-value pairs - it makes "wide" data longer (like `UNPIVOT` in `SQL`)
- `spread`: takes two columns (key & value) and spreads into multiple columns - it makes "long" data wider (like `PIVOT` in `SQL`)

The Data Tidying Operations

Getting data into a tidy format requires understanding the three qualities of tidy data, as well as the five most common types of untidy data (more on these shortly).

Then, we can get most forms of untidy data to be tidy by utilizing four verbs of data tidying

- `gather`: takes multiple columns, and gathers them into key-value pairs - it makes "wide" data longer (like `UNPIVOT` in `SQL`)
- `spread`: takes two columns (key & value) and spreads into multiple columns - it makes "long" data wider (like `PIVOT` in `SQL`)
- `separate`: turns a single column that is character-valued into multiple columns, based on a regular expression or specific positions

The Data Tidying Operations

Getting data into a tidy format requires understanding the three qualities of tidy data, as well as the five most common types of untidy data (more on these shortly).

Then, we can get most forms of untidy data to be tidy by utilizing four verbs of data tidying

- `gather`: takes multiple columns, and gathers them into key-value pairs - it makes "wide" data longer (like `UNPIVOT` in `SQL`)
- `spread`: takes two columns (key & value) and spreads into multiple columns - it makes "long" data wider (like `PIVOT` in `SQL`)
- `separate`: turns a single column that is character-valued into multiple columns, based on a regular expression or specific positions
- `unite`: concatenates multiple columns into one

Execution (the practice)

Five common types of untidy data

Here are the five most common types of untidy data you're likely to experience "in the wild":

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

We'll go through examples of each of the five.

For each untidy dataset...

Think about the following questions:

- Why is it untidy?
- What are the variables, values, and observations?


BRACE YOURSELVES,
UNTIDY DATA IS COMING
imgflip.com

1. Column headers are values, not variable names

The first dataset we'll look at comes from the WHO and displays the number of TB cases for three countries in two years.

country	1999	2000
Afghanistan	745	2,666
Brazil	37,737	80,488
China	212,258	213,766

This data is too *wide*, as 1999 and 2000 are **values** for a **variable** we could call `year`.

Although difficult to analyze, this format is helpful for presentation and data entry.

Tidying # 1

Need to `gather` columns into key-value (year-cases) pairs:

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

2. Multiple variables are stored in one column

The next table has two columns, but it should have four. How would you work with this data without tidying it first?

Hair - Eye - Sex	n
Black - Brown - Male	32
Brown - Brown - Male	53
Red - Brown - Male	10
Blond - Brown - Male	3
Black - Blue - Male	11
Brown - Blue - Male	50

The `Hair - Eye - Sex` **variable** actually has **values** for three separate **variables** stored within it.

Tidying #2

We need to `separate` one column (`Hair - Eye - Sex`) into multiple columns (`hair`, `eye`, `sex`)

hair	eye	sex	n
Black	Brown	Male	32
Brown	Brown	Male	53
Red	Brown	Male	10
Blond	Brown	Male	3
Black	Blue	Male	11
Brown	Blue	Male	50

3. Variables are stored in both rows and columns

This is the most complicated form of untidy data, and typically requires a bit more massaging.

id	year	month	element	d1	d2	d3	d4	d5
MX17004	2010	2	tmax	NA	27.3	24.1	NA	NA
MX17004	2010	2	tmin	NA	14.4	14.4	NA	NA
MX17004	2010	3	tmax	NA	NA	NA	NA	32.1
MX17004	2010	3	tmin	NA	NA	NA	NA	14.2

Think carefully about what the **observation** is for this data.

Tidying #3

blah blah blah

4. Multiple types of observational units are stored in the same table

This is one that gets violated a lot. Our desire is to have *all* the data in one spot.

First: tidy up while normalized

Then: analyze while de-normalized

golfer	birth_date	birth_place	tournament_date	tournament	final_score
Tiger Woods	1975-12-30	Cypress, CA	1996-10-06	Las Vegas	-27
Tiger Woods	1975-12-30	Cypress, CA	1996-10-20	Disney	-21
Tiger Woods	1975-12-30	Cypress, CA	1997-01-12	Mercedes	-14
Tiger Woods	1975-12-30	Cypress, CA	1997-04-13	Masters	-18

5. A single observational unit is stored in multiple tables

Have you ever worked with US government data before?

5. A single observational unit is stored in multiple tables

Have you ever worked with US government data before? If so, you know this is common:

year	cpi
2015	237

year	cpi
2016	240

year	cpi
2017	245

Not hard to remedy, but still annoying and **potentially dangerous**. Easy fix for *consistent* tables: `dplyr::bind_rows`

year	cpi
2015	237
2016	240
2017	245

Displaying and Organizing Data

Displaying tidy data

How can we make it easier to scan raw values in a data table?

- Determine the roles of variables in your analysis (fixed by design of experiment vs. measured during course of experiment)

Displaying tidy data

How can we make it easier to scan raw values in a data table?

- Determine the roles of variables in your analysis (fixed by design of experiment vs. measured during course of experiment)
- Fixed variables should come first, then measured variables

Displaying tidy data

How can we make it easier to scan raw values in a data table?

- Determine the roles of variables in your analysis (fixed by design of experiment vs. measured during course of experiment)
- Fixed variables should come first, then measured variables
    - Order from L-R by degree of fixed-ness. The "most fixed" variables are the key describers of an observation, and are useful when we're trying to scan values.

Displaying tidy data

How can we make it easier to scan raw values in a data table?

- Determine the roles of variables in your analysis (fixed by design of experiment vs. measured during course of experiment)
- Fixed variables should come first, then measured variables
    - Order from L-R by degree of fixed-ness. The "most fixed" variables are the key describers of an observation, and are useful when we're trying to scan values.
- Put related variables next to each other

Displaying tidy data

How can we make it easier to scan raw values in a data table?

- Determine the roles of variables in your analysis (fixed by design of experiment vs. measured during course of experiment)
- Fixed variables should come first, then measured variables
    - Order from L-R by degree of fixed-ness. The "most fixed" variables are the key describers of an observation, and are useful when we're trying to scan values.
- Put related variables next to each other
- Order rows based on the first variable and then break ties with the second and subsequent (fixed) variables after that.

Organizing data in spreadsheets

Broman & Woo (2018) wrote a short paper with 12 tips for organizing data in spreadsheets for sharing, analysis, reproducibility, and collaboration. After reading the tidy data paper, I would recommend reading it.

- Be consistent
  - Codes, NA, names, ID, layout, files, dates, phrases
- Write dates like YYYY-MM-DD
- Do not leave any cells empty
- Put just one thing in a cell
- Organize the data as a single rectangle (with subjects as rows, variables as columns, and with a single header row)
- Create a data dictionary
- Do not include calculations in the raw data files
- Do not use font color or highlighting as data
- Choose good names for things
- Make backups
- Use data validation to avoid data entry errors
- Save the data in plain text files

Conclusion

Four tips to get tidy

1. Put each dataset in a table

Four tips to get tidy

1. Put each dataset in a table
2. Put each variable in a column

Four tips to get tidy

1. Put each dataset in a table
2. Put each variable in a column
3. Ask some questions

Four tips to get tidy

1. Put each dataset in a table
2. Put each variable in a column
3. Ask some questions
    - What are the rows of my dataset (observation, level of detail)?

Four tips to get tidy

1. Put each dataset in a table
2. Put each variable in a column
3. Ask some questions
    - What are the rows of my dataset (observation, level of detail)?
    - Is each column a *distinct* variable?

Four tips to get tidy

1. Put each dataset in a table
2. Put each variable in a column
3. Ask some questions
    - What are the rows of my dataset (observation, level of detail)?
    - Is each column a *distinct* variable?
    - How hard would it be to calculate a grouped aggregation?

Four tips to get tidy

1. Put each dataset in a table
2. Put each variable in a column
3. Ask some questions
    - What are the rows of my dataset (observation, level of detail)?
    - Is each column a *distinct* variable?
    - How hard would it be to calculate a grouped aggregation?
4. Structure and tidy up your data to be manipulated by a computer. Ignore urges to make it easily viewed by a human.

Wrapping up

- Code is for *humans*, data is for *computers* (relevant tweet from Vince Buffalo)

Wrapping up

- Code is for *humans*, data is for *computers* (relevant tweet from Vince Buffalo)
- Be consciously aware of your **values**, **variables**, and **observations**

Wrapping up

- Code is for *humans*, data is for *computers* (relevant tweet from Vince Buffalo)
- Be consciously aware of your **values**, **variables**, and **observations**
- Normalization can be your friend

Wrapping up

- Code is for *humans*, data is for *computers* (relevant tweet from Vince Buffalo)
- Be consciously aware of your **values**, **variables**, and **observations**
- Normalization can be your friend
- Be assertive and understanding

Other resources

There's a bevy of resources I consulted for this presentation. I've arranged these in descending order of importance.

- *The* tidy data paper
- Data Organization in Spreadsheets
- Informal version of tidy data paper
- Practical introduction to tidy data
- Tidy data presentation
- Tidy data analysis (an extension of the tidy data paradigm)
- Tidy Data in Python
- Database Normalization
- Codd's 3rd Normal Form

Here's the link to my GitHub repository.

Questions?

Thanks for listening!