Week 4: 'Tidying' data 2

Joshua Rosenberg, Alex Lishinski

February 11, 2021

Welcome!

Welcome to *week 4*!

Record the meeting

Breakout rooms!

Starting with whomever has the least total letters in their first middle and last names...

Two questions

- What is one particular "messy data" problem have you encountered (or do you anticipate encountering) in your own work?
- What is an example of multiple, separate datasets that you might wish to combine together into a single dataset in your work? If you cannot immediately think of one, consider information that might be recorded or created in two different files, but which could be useful to combine.

(10 minutes)

Review of last week's class

Last week we discussed wrangling and tidying data:

1. Reading in Data
2. Tidying data
3. Our tidy data tools

Homework highlights

Effective and ineffective ways of filtering the data

```
data %>%
  filter(content == 1, content == 2, content == 3, content == 4,
```

```
data %>%
  filter(content == "1", content == "2", content == "3", content
```

```
data %>%
  filter(content == "1" | content == "2" | content == "3" | conte
```

```
data %>% filter(between(content, 1, 5))
```

```
data %>% filter(content %in% c("1", "2", "3", "4", "5"))
```

```
data %>% filter(str_detect(content, "[1-5]"))
```

Homework reminder

Check the output of functions you write carefully; you may find yourself *constantly* looking at and viewing your data!

How do I check/view my data?

There are many ways:

- `my_data` (just typing the name of your data will print info about it!)
- `glance(my_data)`
- `str(my_data)`
- `View(my_data)` (do not include in `.Rmd` document)
- `skimr::skim(my_data)` (must install "skimr" first)
- `psych::describe()` (must install "psych" first)

This week's topics

Overview

1. Tidy data: Data reshaping
2. Working with multiple data frames: joins
3. Grouped data operations with dplyr

We are by no means done with the data tidying tools we discussed last week, so don't feel like you should already have those completely mastered yet.

1. Data reshaping

Outline

What is reshaping?

Why would you reshape?

How do you reshape?

1. Data reshaping

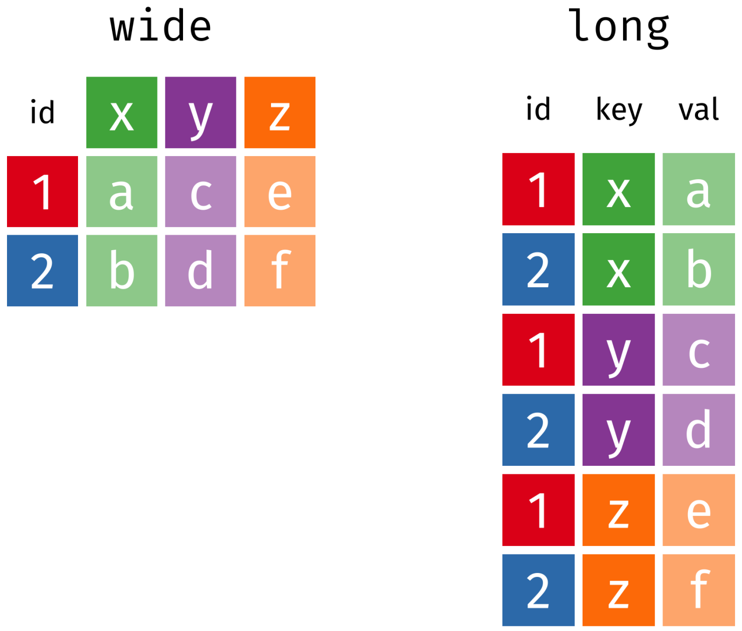
What is reshaping?

- Reshaping involves moving data between "long"er and "wide"er formats
    - Wide data has more columns and fewer rows
    - Long data has more rows and fewer columns
    - This is often a choice you can make with your data

1. Data reshaping

What is reshaping?

- Reshaping involves moving data between "long"er and "wide"er formats



1. Data reshaping

Why would you reshape?

One reason to reshape is to get data in a proper 'tidy' format.

The broader reason is that data can be in multiple shapes and you may need different ones for different purposes.

'Tidy' data doesn't always fully determine what you should do, particularly in the case of repeated measures data.

1. Data reshaping

How do you reshape?

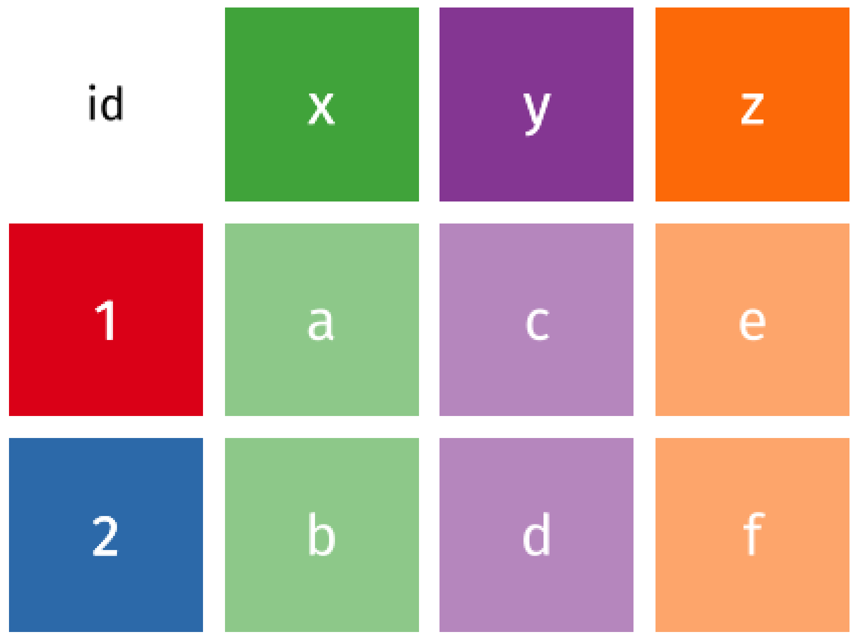
The `tidyr` package offers the `gather()` and `spread()` functions.

These are meant to be replaced by the "pivot" functions from `dplyr`, which are `pivot_longer()` and `pivot_wider()`

1. Data reshaping

How do you reshape?

wide



id	x	y	z
1	a	c	e
2	b	d	f

2. Data joining

What are joins?

Why are joins important?

What are the different types?

How do we know we're doing them right?

2. Data joining

What are joins?

- When you have multiple data frames that you want to combine
- Need to have overlap

Why are joins important?

- They are the way we will use to integrate data from different sources for analysis
- Doing them incorrectly can substantially change your data set

2. Data joining

Different types of joins

- Main types
  - left join
  - right join
  - full join
  - inner join
- Less common
  - semi join
  - anti join

2. Data joining

`dplyr::left_join()`

- Columns from both
- Matching rows from both

2. Data joining

`dplyr::right_join()`

- Columns from both
- Matching rows from both

2. Data joining

`dplyr::full_join()`

- Columns from both
- Matching rows from both

2. Data joining

`dplyr::inner_join()`

- Columns from both
- Matching rows from both

2. Data joining

`dplyr::semi_join()`

- Columns from X
- Matching rows from both

2. Data joining

`dplyr::anti_join()`

- Columns from X
- No matching rows

2. Data joining

How do we know we're doing them right?

- Short answer: Carefully inspecting your data before and after

How do we know which one to choose?

- Short answer: Knowing what the different types do and knowing your data
- It all depends on what you want.

2. Data joining

Matching variables:

- Need to make sure matching variables are correct
- Join functions by default will match all names, but you can specify

Need to align variable names:

- Can do this by renaming variables
- Can also specify corresponding pairs in the join functions

Other issues to consider:

- Multiple matches
- Matching NAs
- Duplicate variable labels

3. Grouped data operations and summary

How do we look at summary information about our variables?

What do we mean by grouping?

How do we group?

What can you do with grouped data?

3. Grouped data operations and summary

How do we look at summary information about our data?

`summarize()` from `dplyr` allows us to look at various sorts of summary info.

3. Grouped data operations and summary

What do we mean by grouping?

Grouping is when we create groups in our data based on a categorical variable (or something that can be transformed into one).

3. Grouped data operations and summary

How do we group?

The `dplyr` function `group_by()` allows us to create grouped data frames.

```r
library(dplyr)

storms %>%
  group_by(name)
```

```
## # A tibble: 5 x 13
## # Groups:   name [1]
##   name   year month   day  hour   lat  long status category  wind pressu
##   <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>  <ord>    <int>    <ir
## 1 Amy    1975     6    27     0  27.5 -79   tropi… -1          25     10
## 2 Amy    1975     6    27     6  28.5 -79   tropi… -1          25     10
## 3 Amy    1975     6    27    12  29.5 -79   tropi… -1          25     10
## 4 Amy    1975     6    27    18  30.5 -79   tropi… -1          25     10
## 5 Amy    1975     6    28     0  31.5 -78.8 tropi… -1          25     10
## # … with 2 more variables: ts_diameter <dbl>, hu_diameter <dbl>
```

3. Grouped data operations and summaries

What can we do with grouped data?

We can combine `summarize()` with grouped data frames and get summary information by group.

We can also do different sorts of data cleaning operations on grouped data.

3. Grouped data operations and summaries

Finding the mean wind speed for storms

```r
library(dplyr)

storms %>%
  group_by(name) %>%
  summarize(mean_wind_speed = mean(wind))
```

```
## # A tibble: 198 x 2
##    name     mean_wind_speed
##  * <chr>              <dbl>
##  1 AL011993            27.5
##  2 AL012000            25  
##  3 AL021992            29  
##  4 AL021994            24.2
##  5 AL021999            28.8
##  6 AL022000            29.2
##  7 AL022001            25  
##  8 AL022003            30  
##  9 AL022006            38  
## 10 AL031987            21.2
## # … with 188 more rows
```

3. Grouped data operations and summaries

Finding the mean wind speed for storms and arranging

```r
library(dplyr)

storms %>%
  group_by(name) %>%
  summarize(mean_wind_speed = mean(wind)) %>%
  arrange(desc(mean_wind_speed))
```

```
## # A tibble: 198 x 2
##    name    mean_wind_speed
##    <chr>             <dbl>
##  1 Wilma              91.9
##  2 Luis               89.2
##  3 Hugo               88.1
##  4 David              86.8
##  5 Gonzalo            86.1
##  6 Ike                83.2
##  7 Igor               81.8
##  8 Joaquin            81.8
##  9 Rita               80.1
## 10 Gilbert            79.6
## # … with 188 more rows
```

Course Logistics

- Exam 1: Available – Tomorrow; Due – Before our next class begins
- Homework 4: (small change; post *one* comment, feedback, or reflections on your homework when you post it to #homework on Slack)
- Personal Data Ethics Statement: Check the assignments page; due 3/4; see assignments details for more information
- Reading on tidying data: https://r4ds.had.co.nz/tidy-data.html

Notes on asking questions

Your #questions are great; keep it up; a few suggestions:

- First, check to see whether someone else asked a question about a similar issue
- When possible and relevant, provide either a) a screen shot or b) a link to your `.Rmd` document along with your question
- Provide as much detail as possible; it's much more helpful to us and others if you explain what's going on, even if you're not certain about terminology
- Consider Googleing the error message.
- If you know the answer, feel free to respond to others!
- If your question is resolved, be sure to post in Slack that you figured it out!

Wrapping up

In your base group's Slack channel:

- What is one thing you took away from today?
- What is something you want to learn more about?