

# Data tidying

From wide-to-long and long-to-wide with `tidyr`

Daniela Palleschi

2024-05-14

## Table of contents

<b>Learning objectives</b>	<b>1</b>
<b>Set-up</b>	<b>2</b>
Inspect data . . . . .	2
<b>Reshape data</b>	<b>3</b>
Pivot with <code>tidyr</code> . . . . .	3
<b>Wide-to-long: <code>pivot_longer()</code></b>	<b>4</b>
<b>Long-to-wide: <code>pivot_wider()</code></b>	<b>5</b>
<b>Re-structuring, not changing</b>	<b>6</b>
<b>Learning objectives</b>	<b>8</b>

## Learning objectives

Today we will...

- learn how to re-structure our data with the `tidyr` package
- use `pivot_longer()` to make data longer
- use `pivot_wider()` to make data wider

## Set-up

Load the `tidyverse` package

```
library(tidyverse)
```

Load a subset of the `tidy_data_lifetime_pilot.csv` data. For demonstration purposes, we'll only look at two trials from a single participant.

```
df_lifetime <- readr::read_csv(here::here("data/tidy_data_lifetime_pilot.csv"),
                              # for special characters
                              locale = readr::locale(encoding = "latin1")
                              ) |>
  filter(type=="critical", px=="px5", trial %in% c(3,8)) |>
  select(px,trial,region,ff,fp,rpd,tt)
```

## Inspect data

- we'll be changing the shape of our data, so let's first see how it looks as-is

```
df_lifetime
```

```
# A tibble: 10 x 7
   px    trial region    ff    fp    rpd    tt
  <chr> <dbl> <chr> <dbl> <dbl> <dbl> <dbl>
1 px5      3 verb-1  190  190  190  190
2 px5      3 verb   175  175  175  321
3 px5      3 verb+1  154  154 1258 1723
4 px5      3 verb+2  160  283  283  672
5 px5      3 verb+3  156  575 1940  575
6 px5      8 verb-1  246  246  246  246
7 px5      8 verb   228  960  960 1892
8 px5      8 verb+1  176  573  573  967
9 px5      8 verb+2  151  151  151  450
10 px5     8 verb+3  216  981 2852  981
```

- of importance, we have the following variables:
  - **region**: contains info on which sentence region the row's reading times correspond to
  - **ff**: first fixation time, an eye-tracking reading measure

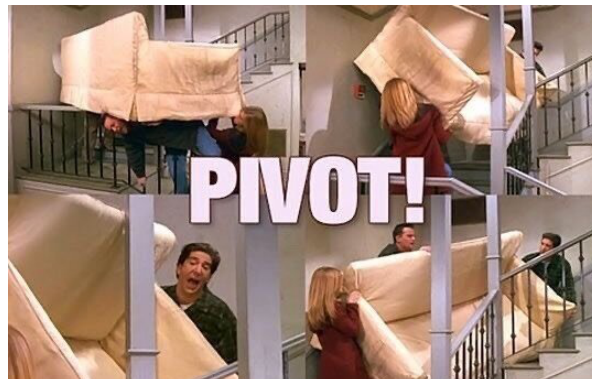
- **fp**: first-pass reading time, an eye-tracking reading measure
- **rpd**: regression path duration, an eye-tracking reading measure
- **tt**: total reading time, an eye-tracking reading measure
- we see that we have 10 rows x 4 reading time measures = 40 reading time measures

## Reshape data

- this is the major step of data tidying
  - make each column a variable
  - make each row an observation
  - make each cell a data point
- what variable and observation mean will depend on what you want to do, and will change at different steps of your analyses
- you typically want *long* data
  - but our dataset isn't as long as it could be
- the **tidyr** package from the **tidyverse** has some useful functions to facilitate this: `pivot_longer()` and `pivot_wider()`

## Pivot with tidyr

- to pivot (verb): *to turn or rotate on a point, like a hinge. Or a basketball player pivoting back and forth on one foot to protect the ball.* ([vocabulary.com](http://vocabulary.com))
- a pivot (noun): *a fixed point supporting something that turns or balances* ([dictionary.Cambridge.org](http://dictionary.cambridge.org))



## Wide-to-long: `pivot_longer()`

- `pivot_longer()` takes wide data and makes it longer
  - converts headers of columns into values of a new column
  - combines the values of those columns into a new condensed column
- takes a few arguments:
  - `cols`: which columns do we want to combine into a single column?
  - `names_to`: what should we call the new column containing the previous column names?
  - `values_to`: what should we call the new column containing the values from the previous columns?
- let's take our four reading time measures and list them in a single variable that we'll call `measure`, and put their values in a second variable called `time`

```
df_longer <-  
df_lifetime |>  
pivot_longer(  
  cols = c(ff,fp,rpd,tt), # columns to make long  
  names_to = "measure", # new column name for headers  
  values_to = "time" # new column name for values  
)
```

```
df_longer
```

```
# A tibble: 40 x 5  
  px    trial region measure  time  
  <chr> <dbl> <chr>   <chr>   <dbl>  
1 px5      3 verb-1 ff      190  
2 px5      3 verb-1 fp      190  
3 px5      3 verb-1 rpd     190  
4 px5      3 verb-1 tt      190  
5 px5      3 verb   ff      175  
6 px5      3 verb   fp      175  
7 px5      3 verb   rpd     175  
8 px5      3 verb   tt      321  
9 px5      3 verb+1 ff      154  
10 px5     3 verb+1 fp      154  
# i 30 more rows
```

- now instead of having the four reading time values in a single row across four columns called `ff`, `fp`, `rp`, and `tt`, we have two columns (`measure` and `time`) which contain the reading time measure names and corresponding reading times
- we again still have 40 reading time values: 40 rows x 1 column containing reading time values (`time`)

## Long-to-wide: `pivot_wider()`

- `pivot_wider()` takes long data and makes it wider
- takes a few arguments:
  - `id_cols`: identifying columns
  - `names_from`: what should we call the new column containing the previous column names?
  - `names_prefix`:
  - `values_from`: new column values
- let's now take our `region` column in `df_longer` and widen it
  - we'll do this only for `tt` (total reading time) the resultfour reading time measures and list them in a single variable that we'll call `measure`, and put their values in a second variable called `time`

```
df_longer_wider <-
  df_longer |>
  pivot_wider(
    id_cols = c(px,trial,measure), # columns to make long
    names_from = region, # new column name for headers
    names_prefix = "reg_", # new column name for values (optional)
    values_from = time
  )
```

```
df_longer_wider
```

```
# A tibble: 8 x 8
  px    trial measure `reg_verb-1` reg_verb `reg_verb+1` `reg_verb+2`
  <chr> <dbl> <chr>         <dbl>   <dbl>         <dbl>         <dbl>
1 px5      3 ff           190     175           154           160
2 px5      3 fp           190     175           154           283
3 px5      3 rpd           190     175          1258           283
4 px5      3 tt           190     321          1723           672
5 px5      8 ff           246     228           176           151
```

```

6 px5      8 fp      246      960      573      151
7 px5      8 rpd     246      960      573      151
8 px5      8 tt      246     1892      967      450
# i 1 more variable: `reg_verb+3` <dbl>

```

- again, we have 40 reading time values: 8 rows x 5 variables containing reading time values per region

## Re-structuring, not changing

- in `df_lifetime`, `df_longer`, and `df_longer_wider`, we have 40 reading time values
  - we have the exact same information in all three versions
  - we have not removed or changed our data
  - we have only changed the *structure* of the data
- this might not always be the case, based on what you're trying to achieve
  - but it's important to understand that you can find the same information in long versus wide data
  - the way you structure your data should reflect/facilitate what you're trying to say about your data
- look at the three versions of the data below, and ask yourself: what does each one more easily communicate?

`df_longer`

```

# A tibble: 40 x 5
  px    trial region measure  time
<chr> <dbl> <chr>   <chr>   <dbl>
1 px5      3 verb-1 ff      190
2 px5      3 verb-1 fp      190
3 px5      3 verb-1 rpd     190
4 px5      3 verb-1 tt      190
5 px5      3 verb   ff      175
6 px5      3 verb   fp      175
7 px5      3 verb   rpd     175
8 px5      3 verb   tt      321
9 px5      3 verb+1 ff      154
10 px5     3 verb+1 fp      154
# i 30 more rows

```

```
# only first 15 rows
df_longer |> head(15)
```

```
# A tibble: 15 x 5
  px    trial region measure  time
  <chr> <dbl> <chr>   <chr>   <dbl>
1 px5      3 verb-1 ff      190
2 px5      3 verb-1 fp      190
3 px5      3 verb-1 rpd     190
4 px5      3 verb-1 tt      190
5 px5      3 verb   ff      175
6 px5      3 verb   fp      175
7 px5      3 verb   rpd     175
8 px5      3 verb   tt      321
9 px5      3 verb+1 ff      154
10 px5     3 verb+1 fp      154
11 px5     3 verb+1 rpd    1258
12 px5     3 verb+1 tt    1723
13 px5     3 verb+2 ff      160
14 px5     3 verb+2 fp      283
15 px5     3 verb+2 rpd     283
```

```
df_longer_wider
```

```
# A tibble: 8 x 8
  px    trial measure `reg_verb-1` reg_verb `reg_verb+1` `reg_verb+2`
  <chr> <dbl> <chr>         <dbl>   <dbl>         <dbl>         <dbl>
1 px5      3 ff          190     175          154          160
2 px5      3 fp          190     175          154          283
3 px5      3 rpd          190     175         1258          283
4 px5      3 tt          190     321         1723          672
5 px5      8 ff          246     228          176          151
6 px5      8 fp          246     960          573          151
7 px5      8 rpd          246     960          573          151
8 px5      8 tt          246    1892          967          450
# i 1 more variable: `reg_verb+3` <dbl>
```

More reading: [PsyTeachR](#)

## Learning objectives

Today we...

- learned how to re-structure our data with the `tidyr` package
- used `pivot_longer()` to make data longer
- used `pivot_wider()` to make data wider