Practice Tidying Data

Anthony Tetreault

2025-03-20

Tidying Data

1. In the following data set, turn the implicit missing values to explicit.

```
output <- tibble(
      treatment   = c("a", "b", "a", "c", "b"),
      gender   = factor(c("M", "F", "F", "M", "M"), levels = c("M", "F", "O")),
      return = c(1.5, 0.75,  0.5,  1.8,  NA)
    )
output %>%
    complete(treatment, gender)
```

```
## # A tibble: 9 x 3
##   treatment gender return
##   <chr>     <fct>   <dbl>
## 1 a         M         1.5
## 2 a         F         0.5
## 3 a         O        NA
## 4 b         M        NA
## 5 b         F         0.75
## 6 b         O        NA
## 7 c         M         1.8
## 8 c         F        NA
## 9 c         O        NA
```

2. Read the dataset available at "https://raw.githubusercontent.com/JaneWall/data_STAT412612/master/weather.csv" as weather.

Use "pivot_longer()" to to put the days all in one column, then use "pivot_wider" to separate tmax and tmin into separate columns. Print the summary of the final resulting dataset.

```
weather <- read_csv("./weather.csv")
```

```
## Rows: 22 Columns: 35
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): id, element
## dbl (25): year, month, d1, d2, d3, d4, d5, d6, d7, d8, d10, d11, d13, d14, d...
## lgl  (8): d9, d12, d18, d19, d20, d21, d22, d24
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
weather %>%
    pivot_longer(
        cols = starts_with("d"),
        names_to = "day",
        names_prefix = "d",
        values_drop_na = TRUE
    ) %>%
        pivot_wider(
            names_from = "element",
            values_from = "value"
        ) %>%
            summary()
```

```
##       id                 year          month            day
##  Length:33          Min.   :2010   Min.   : 1.000   Length:33
##  Class :character   1st Qu.:2010   1st Qu.: 4.000   Class :character
##  Mode  :character   Median :2010   Median : 8.000   Mode  :character
##                     Mean   :2010   Mean   : 7.212
##                     3rd Qu.:2010   3rd Qu.:10.000
##                     Max.   :2010   Max.   :12.000
##       tmax            tmin
##  Min.   :24.10   Min.   : 7.90
##  1st Qu.:27.80   1st Qu.:13.40
##  Median :29.00   Median :15.00
##  Mean   :29.19   Mean   :14.65
##  3rd Qu.:29.90   3rd Qu.:16.50
##  Max.   :36.30   Max.   :18.20
```

3. Tidy the billboard dataset (built-in).

    a. First gather up all the week entries into a row for each week for each song (where there is an entry)
    b. Then, convert the week variable to a number and figure out the date corresponding to each week on the chart
    c. Sort the data by artist, track and week. Here are what your first entries should be (formatting can be different)

```
billboard %>%
    pivot_longer(
        cols = starts_with("wk"),
        names_to = "week",
        names_prefix = "wk",
        names_transform = list(week = as.integer),
        values_drop_na = TRUE
    ) %>%
        mutate(date = date.entered + weeks(week)) %>%
            arrange(artist, track, week)
```

```
## # A tibble: 5,307 x 6
```

```
##    artist  track                   date.entered  week value date
##    <chr>   <chr>                   <date>       <int> <dbl> <date>
##  1 2 Pac   Baby Don't Cry (Keep... 2000-02-26       1    87 2000-03-04
##  2 2 Pac   Baby Don't Cry (Keep... 2000-02-26       2    82 2000-03-11
##  3 2 Pac   Baby Don't Cry (Keep... 2000-02-26       3    72 2000-03-18
##  4 2 Pac   Baby Don't Cry (Keep... 2000-02-26       4    77 2000-03-25
##  5 2 Pac   Baby Don't Cry (Keep... 2000-02-26       5    87 2000-04-01
##  6 2 Pac   Baby Don't Cry (Keep... 2000-02-26       6    94 2000-04-08
##  7 2 Pac   Baby Don't Cry (Keep... 2000-02-26       7    99 2000-04-15
##  8 2Ge+her The Hardest Part Of ... 2000-09-02       1    91 2000-09-09
##  9 2Ge+her The Hardest Part Of ... 2000-09-02       2    87 2000-09-16
## 10 2Ge+her The Hardest Part Of ... 2000-09-02       3    92 2000-09-23
## # i 5,297 more rows
```

4. Load the built in "anscombe" data frame and use "pivot_longer()" to separate all the x and y columns and categorize them into 4 sets.

```
anscombe %>%
     pivot_longer(
         cols = starts_with(c("x","y")),
         names_to = c(".value", "set"),
         names_pattern = "(.)(.)"
     )
```

```
## # A tibble: 44 x 3
##    set       x     y
##    <chr> <dbl> <dbl>
##  1 1        10  8.04
##  2 2        10  9.14
##  3 3        10  7.46
##  4 4         8  6.58
##  5 1         8  6.95
##  6 2         8  8.14
##  7 3         8  6.77
##  8 4         8  5.76
##  9 1        13  7.58
## 10 2        13  8.74
## # i 34 more rows
```

5. As explained in the video load and tidy the built in world_bank_pop data frame.

```
world_bank_pop %>%
    pivot_longer(
        cols = `2000`:`2017`,
        names_to = "year",
        values_to = "value",
        values_drop_na = TRUE
    ) %>%
        separate_wider_regex(
            cols = indicator,
            patterns = c("^.*[:punct:]",
                         area = ".*",
                         "[:punct:]",
```

```
                        variable = ".*$")
        ) %>%
            pivot_wider(
                names_from = "variable",
                values_from = "value"
            )
```

```
## # A tibble: 9,504 x 5
##    country area  year   TOTL   GROW
##    <chr>   <chr> <chr> <dbl>  <dbl>
##  1 ABW     URB   2000  41625 1.66
##  2 ABW     URB   2001  42025 0.956
##  3 ABW     URB   2002  42194 0.401
##  4 ABW     URB   2003  42277 0.197
##  5 ABW     URB   2004  42317 0.0946
##  6 ABW     URB   2005  42399 0.194
##  7 ABW     URB   2006  42555 0.367
##  8 ABW     URB   2007  42729 0.408
##  9 ABW     URB   2008  42906 0.413
## 10 ABW     URB   2009  43079 0.402
## # i 9,494 more rows
```