Practice Tidying Data Lab

Anthony Tetreault

2025-03-20

Libraries Load the tidyverse library

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

Question 1. The following built-in datasets are not tidy. For each one, describe why it is not tidy, write out what the first five entries would look like once it is in a tidy format, and then tidy the dataset

a.relig_income b.billboard c.us_rent_income

```
# relig_income
# This dataset is not tidy because the columns are values of a variable, income, not each variable has
relig_income %>%
    pivot_longer(
        cols = c(`<$10k`, `$10-20k`, `$20-30k`,
                 `$30-40k`, `$40-50k`, `$50-75k`,
                 `$75-100k`, `$100-150k`, `>150k`,
                 `Don't know/refused`),
        names_to = "income",
        values_to = "count"
    )
```

```
## # A tibble: 180 x 3
##    religion income             count
##    <chr>    <chr>              <dbl>
##  1 Agnostic <$10k                 27
##  2 Agnostic $10-20k               34
##  3 Agnostic $20-30k               60
##  4 Agnostic $30-40k               81
##  5 Agnostic $40-50k               76
##  6 Agnostic $50-75k              137
##  7 Agnostic $75-100k             122
##  8 Agnostic $100-150k            109
##  9 Agnostic >150k                 84
## 10 Agnostic Don't know/refused    96
## # i 170 more rows
```

```
# billboard
# This dataset is not tidy because the columns are values of a variable, weeks, not each variable has i
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
```

```
# us_rent_income
# This dataset is not tidy because the column variable contains multiple variables in it (income, rent)
# and each variable should have its own column.
us_rent_income %>%
    pivot_wider(
        id_cols = c("GEOID", "NAME"),
        names_from = "variable",
        values_from = c("estimate", "moe")
    )
```

```
## # A tibble: 52 x 6
##    GEOID NAME                 estimate_income estimate_rent moe_income moe_rent
##    <chr> <chr>                          <dbl>         <dbl>      <dbl>    <dbl>
##  1 01    Alabama                        24476           747        136        3
##  2 02    Alaska                         32940          1200        508       13
##  3 04    Arizona                        27517           972        148        4
##  4 05    Arkansas                       23789           709        165        5
##  5 06    California                     29454          1358        109        3
##  6 08    Colorado                       32401          1125        109        5
##  7 09    Connecticut                       35326          1123         195         5
##  8 10    Delaware                          31560          1076         247        10
##  9 11    District of Columbia              43198          1424         681        17
## 10 12    Florida                           25952          1077          70         3
## # i 42 more rows
```

Question 2. 2.Use "pivot_longer" to tidy the built-in table4b dataset

```
table4b %>%
    pivot_longer(
        cols = c(`1999`, `2000`),
        names_to = "year",
        values_to = "population"
    )
```

```
## # A tibble: 6 x 3
##   country     year  population
##   <chr>       <chr>      <dbl>
## 1 Afghanistan 1999    19987071
## 2 Afghanistan 2000    20595360
## 3 Brazil      1999   172006362
## 4 Brazil      2000   174504898
## 5 China       1999  1272915272
## 6 China       2000  1280428583
```

Question 3. 3.Import and tidy the monkeymen dataset. The cell values represent identification accuracy of some objects (in percent of 20 trials).

```
monkeymen <- read_csv("./monkeymem.csv")
```

```
## Rows: 18 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Monkey, Treatment
## dbl (5): Week2, Week4, Week8, Week12, Week16
##
## 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.
```

```
monkeymen %>%
    pivot_longer( # Collect years into one column, so all columns have one variable
        cols = starts_with("Week"),
        names_to = "Week",
        names_prefix = "Week",
        values_to = "Percent"
    )
```

```
## # A tibble: 90 x 4
##    Monkey Treatment Week  Percent
##    <chr>  <chr>     <chr>   <dbl>
##  1 Spank  Control   2          95
##  2 Spank  Control   4          75
##  3 Spank  Control   8          80
##  4 Spank  Control   12         65
##  5 Spank  Control   16         70
##  6 Chim   Control   2          85
##  7 Chim   Control   4          75
##  8 Chim   Control   8          55
##  9 Chim   Control   12         75
## 10 Chim   Control   16         85
## # i 80 more rows
```

Question 4.

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

```
world_bank_pop %>%
    pivot_longer( # Collect years into one column
        cols = `2000`:`2017`,
        names_to = "year",
        values_to = "value",
        values_drop_na = TRUE
    ) %>%
        separate_wider_regex( # Expand indicator column to capture area and variable
            cols = indicator,
            patterns = c("^.*[:punct:]", # SP.
                         area = ".*", # URB
                         "[:punct:]", # .
                         variable = ".*$") # TOTL/GROW
        ) %>%
            pivot_wider( # Expand variable column to TOTL and GROW columns
                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
```