Homework 5

Carmen Canedo

2 October 2020

Exercise 2.1 Why does tidy data lend itself to vectorised operations?

Tidy data ensures that an observation is always correctly paired with the variables

Exercise 2.2 How could you tidy the SAT data from last week? Which of the data sets below are tidy? What's wrong with the non-tidy data sets?

After reading in the SAT data from the .csv, I placed my data into a tibble. Each variable is in a column, each observation has its own row, and each value has its own cell. To make the SAT data tidy, I would also make all variable names names use snake case (all lowercase letters and underscore instead of spaces), and use title case for all the name of the high schools for consistency.

The only table that is tidy:

- Table 1

The following **are not** tidy:

- Table 2
  - `rate` contains two variables. To fix this we can separate them into `num_cases` and `total_population`, and if we still wanted to include the rate, we could use `mutate()` to divide the two and store the values in `rate`.
- Table 3
  - `2000` and `1999` belong to one variable `year`, but in this table, they are spread across two columns. To fix this, we can use `pivot_longer()` and assign column names to `year` and the values to a separate column, `num_cases`.
- Table 4
  - The observations (country names) in the rows are repeated, so we can use `pivot_wider()` to split `type` into `num_cases` and `total_population`

Exercise 2.3 Use `pivot_longer()` to tidy data frame

```
## # A tibble: 6 x 11
##   religion ‘<$10k‘ ‘$10-20k‘ ‘$20-30k‘ ‘$30-40k‘ ‘$40-50k‘ ‘$50-75k‘ ‘$75-100k‘
##   <chr>      <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>      <dbl>
## 1 Agnostic      27        34        60        81        76       137        122
## 2 Atheist       12        27        37        52        35        70         73
## 3 Buddhist      27        21        30        34        33        58         62
## 4 Catholic     418       617       732       670       638      1116        949
## 5 Don’t k~      15        14        15        11        10        35         21
## 6 Evangel~     575       869      1064       982       881      1486        949
## # ... with 3 more variables: ‘$100-150k‘ <dbl>, ‘>150k‘ <dbl>, ‘Don’t
## #   know/refused‘ <dbl>
```

```
## # 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
## # ... with 170 more rows
```

Exercise 2.4 Tidy the data from blackboard

Exercise 2.5 Use `pivot_wider()` to tidy `tidyr::fishencounters`

Exercise 2.6 Tidy flowers1 data set

Exercise 2.7 Use `separate` to tidy the flowers2 data set

Exercise 2.8 Read the help file for `unite` and correct the code above to get rid of underscore in `year` column

Exercise 2.9 Turn implicit missing values in the data frame

Exercise 2.10 Tidy the `tidyr::billboard` data set

1: Gather up all the week entries into a row for each week for each song where there is an entry

2: Convert the week variable to a number and figure out the date corresponding to each week on the chart

3: Sort the data by artist, track, and week