Data Wrangling Assignment Instructions

Sean Conway

8/4/2021

Data Wrangling Assignment

Today, you will use the skills you have developed to tidy a messy dataset. You will then need to use these tidied data to answer some questions that can only be answered with a clean dataset.

The Data

The data are originally from the US Department of Agriculture. They contain information about the prices of organic food products from 2004-2013[1].

The specific data file we are working with, `eggs_2004_2013.csv`, is a .csv (comma-separated value) file containing a selected portion of data on the price per carton for organic products in the US for each month from 2004-2013.

First, we need to read in the data, using `read_csv()` from the `readr` package (part of the `tidyverse`).

```
eggs <- read_csv(file="eggs_2004_2013.csv")
```

Next, let's take a look at our data.

```
eggs
```

```
## # A tibble: 120 x 6
##    month     year large_half_dozen large_dozen extra_large_half~ extra_large_doz~
##    <chr>    <dbl>            <dbl>       <dbl>             <dbl>            <dbl>
##  1 January   2004             126         230               132              230
##  2 Februa~   2004             128.        226.              134.             230
##  3 March     2004             131         225               137              230
##  4 April     2004             131         225               137              234.
##  5 May       2004             131         225               137              236
##  6 June      2004             134.        231.              137              241
##  7 July      2004             134.        234.              137              241
##  8 August    2004             134.        234.              137              241
##  9 Septem~   2004             130.        234.              136.             241
## 10 October   2004             128.        234.              136.             241
## # ... with 110 more rows
```

[1] Source: https://www.ers.usda.gov/data-products/organic-prices.aspx

```
glimpse(eggs)
```

```
## Rows: 120
## Columns: 6
## $ month                  <chr> "January", "February", "March", "April", "May",~
## $ year                   <dbl> 2004, 2004, 2004, 2004, 2004, 2004, 2004, 2004,~
## $ large_half_dozen       <dbl> 126.00, 128.50, 131.00, 131.00, 131.00, 133.50,~
## $ large_dozen            <dbl> 230.000, 226.250, 225.000, 225.000, 225.000, 23~
## $ extra_large_half_dozen <dbl> 132.000, 134.500, 137.000, 137.000, 137.000, 13~
## $ extra_large_dozen      <dbl> 230.0, 230.0, 230.0, 234.5, 236.0, 241.0, 241.0~
```

These data contain six columns: `month`, `year`, `large_half_dozen`, `large_dozen`, `extra_large_half_dozen`, and `extra_large_dozen`.

Right away, one major problem should be apparent - these data are wide and need to be tidied. Specifically, the size of egg carton is spread across four columns (i.e., `large_half_dozen`, `large_dozen`, `extra_large_half_dozen`, and `extra_large_dozen`). One of your tasks will involve taking these columns and using `pivot_longer()` to tidy the data, creating one column for carton size (`carton_size`) and another for price (`price`).

In addition, the values contained in these four columns are prices per carton. However, you may notice that they look much larger than you see at the grocery store. This is because these values are actually *cents* per carton. You will need to use `mutate()` to create a new column from `price`, called `price_dollar`, where price per carton is converted to a dollar value.

Lastly, while the `month` variable has no particular issues, you will need to use the values of `month` to create a `season` variable. For example, if the value of `month` is `"September"`, `"October"`, or `"November"`, the value of `season` will be `"fall"`.

Assignment Instructions

Here are specific instructions for the assignment. First complete the **Data Wrangling** section. Next, use the tidied data from that section to answer questions in the **Data Questions** section.

Data Wrangling

1. Use `pivot_longer()` to combine the names of the egg carton sizes into a single variable, `carton_size`, while moving the values contained in these columns to another variable, `price`.
2. Use `mutate()` to convert `price` to dollar values, in a new variable called `price_dollar`. Drop `price` from the data.
3. Use `mutate()` and `case_when()` to create a new column, `season`, based on the values of the column `month`. Here are the "rules" for this new column:
    - If `month` is equal to `"September"`, `"October"`, or `"November"`, `season` should have the value `"fall"`.
    - If `month` is equal to `"December"`, `"January"`, or `"February"`, `season` should have the value `"winter"`.
    - If `month` is equal to `"March"`, `"April"`, or `"May"`, `season` should have the value `"spring"`.
    - If `month` is equal to `"June"`, `"July"`, or `"August"`, `season` should have the value `"summer"`.

After the data is tidied, you should be able to use these data (along with `dplyr` verbs like `summarise()`) to answer the next set of questions.

Data Questions

Answer the following questions:

1. How much did a large carton of a half-dozen eggs cost in October 2008?
2. Which month has the highest average price for a large carton of a half-dozen eggs (ignoring the year)?
3. Which year had the highest average price for a an extra large carton of a dozen eggs?
4. In 2009, which season (i.e., fall, winter, spring, summer) had the lowest average price for a large carton of a dozen eggs?
5. What was the median price for one extra-large carton of a half-dozen eggs in summer 2011?