

Eggs Data Wrangling Solution

Sean Conway

8/1/2021

The 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 February  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 September 2004           130.           234.           136.           241
## 10 October  2004           128.           234.           136.           241
## # ... with 110 more rows
```

```
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~
```

DATA WRANGLING - SOLUTION

Creating carton_size & price variables

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`.

Solution Using `pivot_longer()`, we can take our wide dataset and elongate it to make it tidy. First, we define the columns (`cols`) we want, using the colon operator (`:`) to select all four carton size columns. Next, we use `names_to` to take our column names and put them in a single column which we will call `carton_size`. Then, the price values, which were spread across columns, are now placed in a single column, called `price`.

```
eggs_1 <- eggs %>%
  pivot_longer(
    cols=large_half_dozen:extra_large_dozen,
    names_to = "carton_size",
    values_to = "price"
  )
```

Converting price to price_dollar

2. Use `mutate()` to convert `price` to dollar values, in a new variable called `price_dollar`. Drop `price` from the data.

Solution We use `mutate()` to convert `price` to `price_dollars`, and then remove the `price` column (which we no longer need) using `select(-c(price))`.

```
eggs_1_dollar <- eggs_1 %>%
  mutate(
    price_dollar=price/100
  ) %>%
  select(-c(price))
eggs_1_dollar
```

```
## # A tibble: 480 x 4
##   month      year carton_size price_dollar
##   <chr>    <dbl> <chr>          <dbl>
## 1 January  2004 large_half_dozen      1.26
## 2 January  2004 large_dozen        2.3
## 3 January  2004 extra_large_half_dozen 1.32
## 4 January  2004 extra_large_dozen      2.3
## 5 February 2004 large_half_dozen      1.28
## 6 February 2004 large_dozen          2.26
## 7 February 2004 extra_large_half_dozen 1.34
## 8 February 2004 extra_large_dozen      2.3
## 9 March    2004 large_half_dozen      1.31
## 10 March   2004 large_dozen          2.25
## # ... with 470 more rows
```

Creating a season variable

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".

Solution First, we need to create a new variable, called `season`, which classifies each month as being in winter, spring, summer, or fall. There are two solutions here (though other solutions are possible).

Solution 1 - The Long Way We use `mutate()` and `case_when()` to create a variable called `season`, where the value of `month` is used to classify each row as winter, spring, summer, or fall.

```
eggs_tidy <- eggs_1_dollar %>%
  mutate(
    season=case_when(
      month == "December" | month == "January" | month == "February" ~ "winter",
      month == "March" | month == "April" | month == "May" ~ "spring",
      month == "June" | month == "July" | month == "August" ~ "summer",
      month == "September" | month == "October" | month == "November" ~ "fall"
    )
  )
```

Solution 1 - The results

```
eggs_tidy

## # A tibble: 480 x 5
##   month      year carton_size      price_dollar season
##   <chr>    <dbl> <chr>          <dbl> <chr>
## 1 January  2004 large_half_dozen      1.26 winter
## 2 January  2004 large_dozen          2.3  winter
## 3 January  2004 extra_large_half_dozen 1.32 winter
## 4 January  2004 extra_large_dozen        2.3  winter
## 5 February 2004 large_half_dozen      1.28 winter
## 6 February 2004 large_dozen          2.26 winter
## 7 February 2004 extra_large_half_dozen 1.34 winter
## 8 February 2004 extra_large_dozen        2.3  winter
## 9 March    2004 large_half_dozen      1.31 spring
## 10 March   2004 large_dozen          2.25 spring
## # ... with 470 more rows
```

Solution 2 - The Short(er) Way We can first create four vectors, `winter`, `spring`, `summer`, `fall`, each containing the names of the months in the particular season.

Then, we use `mutate()`, `case_when()`, and the `%in%` operator to check if each row's `month` value is in each vector. For example, if the `month` in a particular row is "January", the `case_when()` call will return a value of `TRUE`, and call it `winter`. This continues for each of the four seasons.

```
winter <- c("December", "January", "February")
spring <- c("March", "April", "May")
summer <- c("June", "July", "August")
fall <- c("September", "October", "November")

eggs_tidy_1 <- eggs_1 %>%
  mutate(
    season=case_when(
      month %in% winter ~ "winter",
      month %in% spring ~ "spring",
```

```

    month %in% summer ~ "summer",
    month %in% fall ~ "fall"
  )
)

```

Solution 2 - The results

```
eggs_tidy_1
```

```

## # A tibble: 480 x 5
##   month      year carton_size      price season
##   <chr>    <dbl> <chr>          <dbl> <chr>
## 1 January    2004 large_half_dozen    126  winter
## 2 January    2004 large_dozen        230  winter
## 3 January    2004 extra_large_half_dozen  132  winter
## 4 January    2004 extra_large_dozen    230  winter
## 5 February   2004 large_half_dozen    128.  winter
## 6 February   2004 large_dozen        226.  winter
## 7 February   2004 extra_large_half_dozen  134.  winter
## 8 February   2004 extra_large_dozen    230  winter
## 9 March      2004 large_half_dozen    131  spring
## 10 March     2004 large_dozen        225  spring
## # ... with 470 more rows

```

The results look the same, so we'll just use `eggs_tidy` for the remainder of the file.

DATA QUESTIONS - SOLUTION

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?

Question 1 - Solution

1. How much did a large carton of a half-dozen eggs cost in October 2008?

```

eggs_tidy %>%
  filter(month=="October" & year==2004 & carton_size=="large_half_dozen")

```

```

## # A tibble: 1 x 5
##   month      year carton_size      price_dollar season
##   <chr>    <dbl> <chr>          <dbl> <chr>
## 1 October    2004 large_half_dozen    1.28 fall

```

The answer is \$1.28.

Question 2 - Solution

2. Which month has the highest average price for a large carton of a half-dozen eggs (ignoring the year)?

```
eggs_tidy %>%
  filter(carton_size=="large_half_dozen") %>%
  group_by(month) %>%
  summarise(avg_price=mean(price_dollar)) %>%
  arrange(desc(avg_price))
```

```
## # A tibble: 12 x 2
##   month      avg_price
##   <chr>      <dbl>
## 1 August      1.57
## 2 July        1.57
## 3 June        1.57
## 4 December   1.57
## 5 November   1.57
## 6 September   1.57
## 7 October    1.57
## 8 April       1.53
## 9 March       1.53
## 10 May        1.53
## 11 February   1.52
## 12 January    1.52
```

It's a tie! between 7 months - June, July, August, December, September, October, November, with an average price of \$1.57.

Question 3 - Solution

3. Which year had the highest average price for a an extra large carton of a dozen eggs?

```
eggs_tidy %>%
  filter(carton_size=="extra_large_dozen") %>%
  group_by(year) %>%
  summarise(avg_price=mean(price_dollar)) %>%
  arrange(desc(avg_price))
```

```
## # A tibble: 10 x 2
##   year avg_price
##   <dbl>   <dbl>
## 1 2013      2.9
## 2 2012     2.88
## 3 2009     2.86
## 4 2010     2.86
## 5 2011     2.86
## 6 2008     2.69
## 7 2007     2.45
## 8 2006     2.41
## 9 2005     2.41
## 10 2004     2.37
```

2013 had the highest average price for extra large dozen eggs, at \$2.90.

Question 4 - Solution

4. In 2009, which season (i.e., fall, winter, spring, summer) had the lowest average price for a large carton of a dozen eggs?

```
eggs_tidy %>%
  filter(carton_size=="large_dozen" & year==2009) %>%
  group_by(season) %>%
  summarise(avg_price=mean(price_dollar)) %>%
  arrange(desc(avg_price))
```

```
## # A tibble: 4 x 2
##   season avg_price
##   <chr>     <dbl>
## 1 spring     2.78
## 2 summer     2.76
## 3 winter     2.76
## 4 fall       2.72
```

Spring is the answer, with an average price of \$2.78.

Question 5 - Solution

5. What was the median price for one extra-large carton of a half-dozen eggs in summer 2011?

```
eggs_tidy %>%
  filter(carton_size=="extra_large_half_dozen" & year==2004 & season=="summer") %>%
  summarise(med_price=median(price_dollar))
```

```
## # A tibble: 1 x 1
##   med_price
##   <dbl>
## 1       1.37
```

The answer is \$1.37.