

For the letter C, we'll talk about the coalesce function. If you're familiar with SQL, you may have seen this function before. It combines two or more variables into a single column, and is a way to deal with missing data. When you give it a list of variables, it selects the first non-missing value it finds. Because of that, order of entry of those variables is important.

I recently used this function at work to get last performance on an item. That is, diplomates in our program take spaced assessments, with repetition of items (same content, but slightly different item wording) that they answered incorrectly. If they got the initial item correct, they may never see a related item again (although over time they will), but if they were incorrect, they may get related items once or more times to see if they're learning. So we have some people who will have a missing value for the repeated item and others who will have correct/incorrect. The coalesce function allowed me to get their final performance on item content by telling it to first check the repeated item column and then, if that's missing, to look at the initial item column.

How can we use the coalesce function for the reading dataset? I noticed that for 17 books in the dataset, I forgot to supply a rating, so they're showing up as 0. Let's select those titles, and I can figure out how I'd like to rate them.

```
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.0 --

##  ggplot2 3.2.1      purrr   0.3.3
##  tibble  2.1.3      dplyr   0.8.3
##  tidyr   1.0.0      stringr 1.4.0
##  readr   1.3.1      forcats 0.4.0

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

reads2019 <- read_csv("~/Downloads/Blogging A to Z/SarasReads2019.csv", col_names = TRUE)

## Parsed with column specification:
## cols(
##   Title = col_character(),
##   Pages = col_double(),
##   date_started = col_character(),
##   date_read = col_character(),
##   Book.ID = col_double(),
##   Author = col_character(),
##   AdditionalAuthors = col_character(),
##   AverageRating = col_double(),
##   OriginalPublicationYear = col_double(),
##   read_time = col_double(),
##   MyRating = col_double(),
##   Gender = col_double(),
##   Fiction = col_double(),
##   Childrens = col_double(),
##   Fantasy = col_double(),
##   SciFi = col_double(),
##   Mystery = col_double(),
##   SelfHelp = col_double()
## )

unrated <- reads2019 %>%
  filter(MyRating == 0)

list(unrated$Title)
```

```
## [[1]]
## [1] "Bird Box (Bird Box, #1)"
## [2] "Elevation"
## [3] "Glinda of Oz (Oz, #14)"
## [4] "It"
## [5] "Just Evil (Evil Secrets Trilogy, #1)"
## [6] "Oryx and Crake (MaddAddam, #1)"
## [7] "Precarious Pasta (Cozy Corgi Mysteries, #14)"
## [8] "Redshirts"
## [9] "Rinkitink in Oz (Oz, #10)"
## [10] "Summerdale"
## [11] "The 5 Love Languages: The Secret to Love That Lasts"
## [12] "The Long Mars (The Long Earth, #3)"
## [13] "The Magic of Oz (Oz, #13)"
## [14] "The Marvelous Land of Oz (Oz, #2)"
## [15] "The Scarecrow of Oz (Oz, #9)"
## [16] "The Tin Woodman of Oz (Oz, #12)"
## [17] "Tik-Tok of Oz (Oz, #8)"
```

So now I want to generate a variable with my new ratings, and merge it into my reads dataset.

```
unrated$NewRating <- c(3, 5, 4, 4, 2, 4, 4, 5, 4, 3, 3, 4, 4, 5, 3, 3, 4)
```

```
unrated <- unrated %>%
  select(Book.ID, NewRating)
```

```
reads2019 <- reads2019 %>%
  left_join(unrated, by = "Book.ID")
```

Finally, we'll create our final ratings variable by coalescing the two ratings variables. I want it to first look at the new rating column, so it can capture those updates. For rows missing that variable (because I already rated them), it will look next at the original rating variable.

```
reads2019 <- reads2019 %>%
  mutate(FinalRating = coalesce(NewRating, MyRating))
```

```
table(reads2019$FinalRating)
```

```
##
##  2  3  4  5
##  2  9 51 25
```

No more 0s!