

Moving on to the letter B, today we'll talk about merging datasets that contain the same variables but add new cases. This is easily done with `bind_rows`. Let's say I realized I forgot to log some of the books I read last year, and I wanted to merge those in to my existing dataset. I selected a handful of books from my to-read list, generated some read time and rating data, and saved the results in a csv file (which you can find [here](#)). Now I want to load my existing dataset and the new one:

```
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()
## )

addreads <- read_csv("~/Downloads/Blogging A to Z/SarasAdds.csv")

## 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()
## )
```

Now we just bind the two datasets together:

```
reads2019 <- reads2019 %>%
  bind_rows(addreads)
```

Did these additions change the ordering by page length?

```
reads2019 <- reads2019 %>%
  arrange(desc(Pages), Author)
```

```
head(reads2019)
```

```
## # A tibble: 6 x 18
##   Title Pages date_started date_read Book.ID Author AdditionalAutho...
##
## 1 The ... 1216 6/12/2019    6/18/2019 3.30e1 Tolki...
## 2 The ... 1181 6/12/2019    6/17/2019 1.86e7 Atwoo...
## 3 It      1156 8/14/2019    8/21/2019 2.79e7 King,...
## 4 1Q84     925 9/3/2019     9/10/2019 1.04e7 Murak... Jay Rubin, Phil...
## 5 Inso...  890 8/10/2019    8/13/2019 1.06e4 King,... Bettina Blanch ...
## 6 The ...  592 8/18/2019    8/23/2019 1.16e4 King,...
## # ... with 11 more variables: AverageRating , OriginalPublicationYear ,
## #   read_time , MyRating , Gender , Fiction ,
## #   Childrens , Fantasy , SciFi , Mystery , SelfHelp
```

It did! The longest book is now *The Lord of the Rings*, at 1216 pages, and number two is *The MaddAddam Trilogy*, 1181 pages.

This is a pretty easy trick. Later on in this series, we'll talk about combining datasets that share cases but add new variables – joins – which is one of the times the tidy data mindset becomes very important.