A few times in this series, I've wanted to display part of a dataset, such as key variables, like Title, Rating, and Pages. The tidyverse allows you to easily keep or drop variables, either temporarily or permanently, with the select function. For instance, we can use select along with other tidyverse functions to create a quick descriptive table of my dataset. Let's filter down to books that are fantasy and/or sci-fi and that took me the longest to read, then select a few descriptives to display.

```
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/SaraReads2019_allrated.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()
## )
```

```
reads2019 %>%
  group_by(Fantasy, SciFi) %>%
  filter(read_time == max(read_time) & (Fantasy == 1 | SciFi == 1)) %>%
  select(Title, Author, Pages, read_time)
```

```
## Adding missing grouping variables: `Fantasy`, `SciFi`
```

```
## # A tibble: 4 x 6
## # Groups:   Fantasy, SciFi [3]
##   Fantasy SciFi Title                              Author        Pages
read_time
##
```

```
## 1       1     1 1Q84                                Murakami, Ha…   925
7
## 2       0     1 The End of All Things (Old Man's … Scalzi, John    380
10
## 3       0     1 The Long Utopia (The Long Earth #… Pratchett, T…   373
10
## 4       1     0 Tik-Tok of Oz (Oz, #8)             Baum, L. Fra…   272
25
```

Of course, I can also permanently change the reads2019 dataset to only keep those variables or create a new dataset with just those variables. The select function can also be used to drop single variables, by putting a – sign before the variable name. Let's say I decided I no longer wanted to keep the Self Help genre flag. I could throw that out of my dataset like this.

```
reads2019 <- reads2019 %>%
  select(-SelfHelp)
```

That variable is now gone. You can use this same code to drop multiple variables at once, by putting – signs before each variable name.

```
small_reads2019 <- reads2019 %>%
  select(-AdditionalAuthors, -AverageRating, -OriginalPublicationYear)
```

Whichever you do, keeping or dropping, choose the option that minimizes how many things you have to type. If you have a large number of variables and want a dataset with only a handful, I'd use the names of the variables I want to keep with select. If you only want to drop 1 or 2 variables, using select to drop will be faster.