

```

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()
## )

```

Using this dataset, I'm going create and modify multiple variables – I'll turn some of my genre flags into factors, which aids in data visualization; convert my start and finished dates from character to date variables; extract the day of the week I started and finished each book; and label books based on whether they were recently published (in the last five years) or older.

```

library(lubridate)

##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##   date

reads2019 <- reads2019 %>%
  mutate(Fiction = factor(Fiction,
                           levels = c(0,1),
                           labels = c("Non-Fiction", "Fiction")),
         Fantasy = factor(Fantasy, levels = c(0,1),

```

```

      labels = c("Non-Fantasy", "Fantasy")),
SciFi = factor(SciFi,
      levels = c(0,1),
      labels = c("Non-Science Fiction", "Science Fiction")),
date_started = as.Date(reads2019$date_started, format = '%m/%d/%Y'),
date_read = as.Date(date_read, format = '%m/%d/%Y'),
StartDay = wday(date_started, label = TRUE, abbr = FALSE),
FinishDay = wday(date_read, label = TRUE, abbr = FALSE),
Age = ifelse(OriginalPublicationYear >= 2015, 1, 0),
Age = factor(Age,
      levels = c(0,1),
      labels = c("New Publication", "Older Publication"))

```

Notice that I both created the Age variable and turned it into a factor, so it's okay to use a newly created variable in the same mutate wrap. I also converted my character dates into date format so that I could go on to extract day of the week (wday, a function from lubridate). In fact, this block of mutate code includes both base R and tidyverse functions. I could even pull in functions from other R packages. Basically any code you used to write as `data$variable <- f(data$variable)`, you can embed in mutate as:

```

data <- data %>%
  mutate(variable = f(variable))

```

Sure, it seems more verbose when you're doing one quick change, but changing datasets often involve multiple variables, and may require you to use mutate with filters or group\_bys. That makes tidyverse much more powerful and cleaner.