Dplyr

Mathew Katz

2022-11-12

When working with data you must:

- Figure out what you want to do.
- Describe those tasks in the form of a computer program.
- Execute the program.

The dplyr package makes these steps fast and easy:

- By constraining your options, it helps you think about your data manipulation challenges.
- It provides simple "verbs", functions that correspond to the most common data manipulation tasks, to help you translate your thoughts into code.
- It uses efficient backends, so you spend less time waiting for the computer.

Let's load in dplyr through Tidyverse:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.0      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Import our data:

```
df <- read_csv('bestsellers.csv', show_col_types = FALSE)
```

Best way to get an idea of what your data is/looks like:

```
#Glimpse is like a transposed version of print(): columns run down the page, and data runs across. This
glimpse(df)
```

```
## Rows: 550
## Columns: 7
## $ Name          <chr> "10-Day Green Smoothie Cleanse", "11/22/63: A Novel", "1~
## $ Author        <chr> "JJ Smith", "Stephen King", "Jordan B. Peterson", "Georg~
## $ `User Rating` <dbl> 4.7, 4.6, 4.7, 4.7, 4.8, 4.4, 4.7, 4.7, 4.7, 4.6, 4.6, 4~
## $ Reviews       <dbl> 17350, 2052, 18979, 21424, 7665, 12643, 19735, 19699, 59~
## $ Price         <dbl> 8, 22, 15, 6, 12, 11, 30, 15, 3, 8, 8, 2, 32, 5, 17, 4, ~
## $ Year          <dbl> 2016, 2011, 2018, 2017, 2019, 2011, 2014, 2017, 2018, 20~
## $ Genre         <chr> "Non Fiction", "Fiction", "Non Fiction", "Fiction", "Non~
```

```
#Returns the first parts of a vector, matrix, table, data frame or function.
head(df)
```

```
## # A tibble: 6 x 7
##   Name                                Author User ~1 Reviews Price  Year Genre
##   <chr>                               <chr>    <dbl>   <dbl> <dbl> <dbl> <chr>
## 1 10-Day Green Smoothie Cleanse       JJ Sm~     4.7   17350     8  2016 Non ~
## 2 11/22/63: A Novel                   Steph~     4.6    2052    22  2011 Fict~
## 3 12 Rules for Life: An Antidote to Ch~ Jorda~     4.7   18979    15  2018 Non ~
## 4 1984 (Signet Classics)              Georg~     4.7   21424     6  2017 Fict~
## 5 5,000 Awesome Facts (About Everythin~ Natio~     4.8    7665    12  2019 Non ~
## 6 A Dance with Dragons (A Song of Ice ~ Georg~     4.4   12643    11  2011 Fict~
## # ... with abbreviated variable name 1: `User Rating`
```

```
#Generic function used to produce result summaries of the results of various model fitting functions
summary(df)
```

```
##      Name              Author           User Rating       Reviews
##  Length:550         Length:550         Min.   :3.300   Min.   :   37
##  Class :character   Class :character   1st Qu.:4.500   1st Qu.: 4058
##  Mode  :character   Mode  :character   Median :4.700   Median : 8580
##                                        Mean   :4.618   Mean   :11953
##                                        3rd Qu.:4.800   3rd Qu.:17253
##                                        Max.   :4.900   Max.   :87841
##      Price             Year         Genre
##  Min.   :  0.0   Min.   :2009   Length:550
##  1st Qu.:  7.0   1st Qu.:2011   Class :character
##  Median : 11.0   Median :2014   Mode  :character
##  Mean   : 13.1   Mean   :2014
##  3rd Qu.: 16.0   3rd Qu.:2017
##  Max.   :105.0   Max.   :2019
```

Find out your column names:

```
names(df)
```

```
## [1] "Name"        "Author"      "User Rating" "Reviews"     "Price"
## [6] "Year"        "Genre"
```

Data Exploration:

```
#function used to subset a data frame, retaining all rows that satisfy your conditions
df <- df %>%
  filter(Reviews >= 10000)
```

Now we only have to look at the books that have significant book reviews making our dataset change from 550 to 225 books.

What is that %>%? It's called a pipe. All of the dplyr functions take a data frame as the first argument. Rather than forcing the user to either save intermediate objects or nest functions, dplyr provides the %>% operator from magrittr. x %>% f(y) turns into f(x, y) so the result from one step is then "piped" into the next step. You can use the pipe to rewrite multiple operations that you can read left-to-right, top-to-bottom (reading the pipe operator as "then").

```
#orders the rows of a data frame by the values of selected columns
df %>%
  arrange(desc(`User Rating`)) %>%
  head()
```

```
## # A tibble: 6 x 7
##   Name                                  Author User ~1 Reviews Price  Year Genre
##   <chr>                                 <chr>    <dbl>   <dbl> <dbl> <dbl> <chr>
## 1 Brown Bear, Brown Bear, What Do You ~ Bill ~     4.9   14344     5  2017 Fict~
## 2 Brown Bear, Brown Bear, What Do You ~ Bill ~     4.9   14344     5  2019 Fict~
## 3 Dog Man: Fetch-22: From the Creator ~ Dav P~     4.9   12619     8  2019 Fict~
## 4 Harry Potter and the Chamber of Secr~ J.K. ~     4.9   19622    30  2016 Fict~
## 5 Harry Potter and the Sorcerer's Ston~ J.K. ~     4.9   10052    22  2016 Fict~
## 6 Jesus Calling: Enjoying Peace in His~ Sarah~     4.9   19576     8  2011 Non ~
## # ... with abbreviated variable name 1: 'User Rating'
```

```
#Select variables in a data frame, using a concise mini-language that makes it easy to refer to variabl
df %>%
  select(Name, Author, Genre)
```

```
## # A tibble: 225 x 3
##    Name                                                          Author Genre
##    <chr>                                                         <chr>  <chr>
##  1 10-Day Green Smoothie Cleanse                                 JJ Sm~ Non ~
##  2 12 Rules for Life: An Antidote to Chaos                       Jorda~ Non ~
##  3 1984 (Signet Classics)                                        Georg~ Fict~
##  4 A Dance with Dragons (A Song of Ice and Fire)                 Georg~ Fict~
##  5 A Game of Thrones / A Clash of Kings / A Storm of Swords / A Fe~ Georg~ Fict~
##  6 A Gentleman in Moscow: A Novel                                Amor ~ Fict~
##  7 A Man Called Ove: A Novel                                     Fredr~ Fict~
##  8 A Man Called Ove: A Novel                                     Fredr~ Fict~
##  9 All the Light We Cannot See                                   Antho~ Fict~
## 10 All the Light We Cannot See                                   Antho~ Fict~
## # ... with 215 more rows
```

```
df %>%
  select(where(is.character))
```

```
## # A tibble: 225 x 3
```

```
##    Name                                                              Author Genre
##    <chr>                                                             <chr>  <chr>
##  1 10-Day Green Smoothie Cleanse                                     JJ Sm~ Non ~
##  2 12 Rules for Life: An Antidote to Chaos                           Jorda~ Non ~
##  3 1984 (Signet Classics)                                            Georg~ Fict~
##  4 A Dance with Dragons (A Song of Ice and Fire)                     Georg~ Fict~
##  5 A Game of Thrones / A Clash of Kings / A Storm of Swords / A Fe~ Georg~ Fict~
##  6 A Gentleman in Moscow: A Novel                                    Amor ~ Fict~
##  7 A Man Called Ove: A Novel                                         Fredr~ Fict~
##  8 A Man Called Ove: A Novel                                         Fredr~ Fict~
##  9 All the Light We Cannot See                                       Antho~ Fict~
## 10 All the Light We Cannot See                                       Antho~ Fict~
## # ... with 215 more rows
```

```
#adds new variables and preserves existing ones
#Convert Book price from dollar to euro
df %>%
  mutate(Price = Price * 0.96)
```

```
## # A tibble: 225 x 7
##    Name                                Author User ~1 Reviews Price  Year Genre
##    <chr>                               <chr>    <dbl>   <dbl> <dbl> <dbl> <chr>
##  1 10-Day Green Smoothie Cleanse       JJ Sm~     4.7   17350  7.68  2016 Non ~
##  2 12 Rules for Life: An Antidote to C~ Jorda~     4.7   18979 14.4   2018 Non ~
##  3 1984 (Signet Classics)              Georg~     4.7   21424  5.76  2017 Fict~
##  4 A Dance with Dragons (A Song of Ice~ Georg~     4.4   12643 10.6   2011 Fict~
##  5 A Game of Thrones / A Clash of King~ Georg~     4.7   19735 28.8   2014 Fict~
##  6 A Gentleman in Moscow: A Novel      Amor ~     4.7   19699 14.4   2017 Fict~
##  7 A Man Called Ove: A Novel           Fredr~     4.6   23848  7.68  2016 Fict~
##  8 A Man Called Ove: A Novel           Fredr~     4.6   23848  7.68  2017 Fict~
##  9 All the Light We Cannot See         Antho~     4.6   36348 13.4   2014 Fict~
## 10 All the Light We Cannot See         Antho~     4.6   36348 13.4   2015 Fict~
## # ... with 215 more rows, and abbreviated variable name 1: `User Rating`
```

```
#creates a new data frame.It will contain one column for each grouping variable and one column for each
df %>%
  summarise(AVG_Price_In_Euros = mean(Price))
```

```
## # A tibble: 1 x 1
##   AVG_Price_In_Euros
##                <dbl>
## 1               10.9
```

```
# takes an existing tbl and converts it into a grouped tbl where operations are performed "by group".
df %>%
  group_by(Year) %>%
  summarise(AVG_Price_In_Euros = mean(Price))
```

```
## # A tibble: 11 x 2
##     Year AVG_Price_In_Euros
##    <dbl>              <dbl>
##  1  2009               9
```

```
##  2  2010             10.9
##  3  2011             12.2
##  4  2012             14.2
##  5  2013             10.4
##  6  2014             11.7
##  7  2015             10.0
##  8  2016             11.5
##  9  2017              9.81
## 10  2018              9.69
## 11  2019              9.78
```