For the letter F – filters! Filters are incredibly useful, especially when combined with the main pipe %>%. I frequently use filters along with ggplot functions, to chart a specific subgroup or remove missing cases or outliers. As one example, I could use a filter to chart only fiction books from my reading dataset.

```
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
```

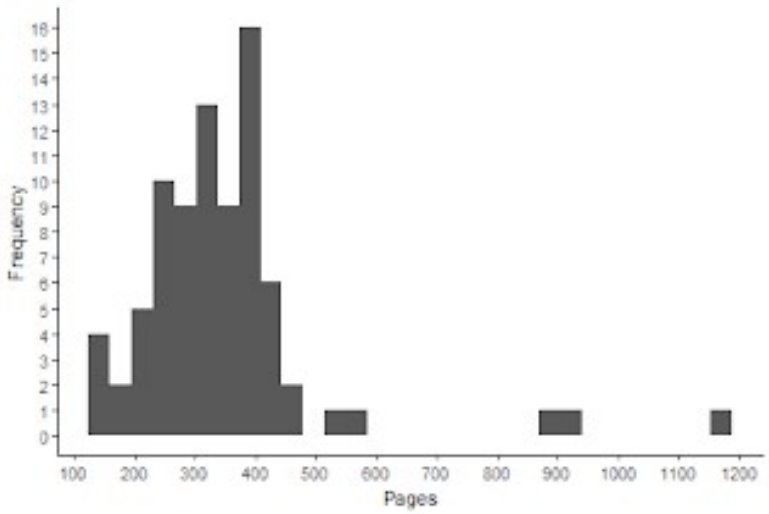
```
## -- Conflicts ------------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
reads2019 <- read_csv("~/Downloads/Blogging A to Z/SarasReads2019_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 %>%
  filter(Fiction == 1) %>%
  ggplot(aes(Pages)) +
  geom_histogram() +
  scale_y_continuous(breaks = seq(0,16,1)) +
  scale_x_continuous(breaks = seq(0,1200,100)) +
  ylab("Frequency") +
  theme_classic()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



I could also use filters to create a new dataset – perhaps one of my top books I read during 2019.

```
library(magrittr)
```

```
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:purrr':
##
##     set_names
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
top_books <- reads2019 %>%
  filter(MyRating == 5)

top_books %$%
  list(Title)
```

```
## [[1]]
##  [1] "1Q84"
##  [2] "Alas, Babylon"
##  [3] "Elevation"
##  [4] "Guards! Guards! (Discworld, #8; City Watch #1)"
##  [5] "How Music Works"
##  [6] "Lords and Ladies (Discworld, #14; Witches #4)"
##  [7] "Moving Pictures (Discworld, #10; Industrial Revolution, #1)"
##  [8] "Redshirts"
##  [9] "Swarm Theory"
## [10] "The Android's Dream (The Android's Dream #1)"
## [11] "The Dutch House"
## [12] "The Emerald City of Oz (Oz #6)"
## [13] "The End of Mr. Y"
## [14] "The Human Division (Old Man's War, #5)"
## [15] "The Last Colony (Old Man's War, #3)"
## [16] "The Long Utopia (The Long Earth #4)"
## [17] "The Marvelous Land of Oz (Oz, #2)"
## [18] "The Miraculous Journey of Edward Tulane"
## [19] "The Night Circus"
## [20] "The Patchwork Girl of Oz (Oz, #7)"
## [21] "The Patron Saint of Liars"
```

```
## [22] "The Wonderful Wizard of Oz (Oz, #1)"
## [23] "The Year of the Flood (MaddAddam, #2)"
## [24] "Witches Abroad (Discworld, #12; Witches #3)"
## [25] "Wyrd Sisters (Discworld, #6; Witches #2)"
```

Or I could create one of the 10 longest books I read:

```
long_books <- reads2019 %>%
  arrange(desc(Pages)) %>%
  filter(between(row_number(), 1, 10)) %>%
  select(Title, Pages)

library(expss)

##
## Use 'expss_output_viewer()' to display tables in the RStudio Viewer.
##  To return to the console output, use 'expss_output_default()'.

##
## Attaching package: 'expss'

## The following objects are masked from 'package:magrittr':
##
##     and, equals, or

## The following objects are masked from 'package:stringr':
##
##     fixed, regex

## The following objects are masked from 'package:dplyr':
##
##     between, compute, contains, first, last, na_if, recode, vars

## The following objects are masked from 'package:purrr':
##
##     keep, modify, modify_if, transpose

## The following objects are masked from 'package:tidyr':
##
##     contains, nest

## The following object is masked from 'package:ggplot2':
##
##     vars

as.etable(long_books, rownames_as_row_labels = FALSE)
```

Title	Pages
It	1156
1Q84	925
Insomnia	890
The Institute	576
The Robber Bride	528
Life of Pi	460
Cell	449
Cujo	432
The Human Division (Old Man’s War, #5)	431
The Year of the Flood (MaddAddam, #2)	431

I can also filter on multiple criteria, with logical operators. To filter on two things, I’d combine them with &. In

this example, I'll select the books that took me longer than a week to read and that were at least 400 pages long.

```
reads2019 %>%
  filter(read_time > 7 & Pages >= 400) %>%
  select(Title, Pages, Author, read_time)
```

```
## # A tibble: 2 x 4
##   Title                              Pages Author           read_time
##
## 1 The Long War (The Long Earth, #2)   419 Pratchett, Terry         8
## 2 The Robber Bride                    528 Atwood, Margaret         9
```

Lastly, let's filter with "or", so we select cases that meet one of the two criteria. We create or with | . The first criteria is read time less than 1 day (meaning I started and finished the book in the same day). The second criteria are my long reads/long books criteria from above. Since there's two parts to this side of the |, I enclose them in parentheses so the statement is evaluated together across the data:

```
reads2019 %>%
  filter(read_time < 1 | (read_time > 7 & Pages >= 400)) %>%
  select(Title, Pages, Author, read_time)
```

```
## # A tibble: 4 x 4
##   Title                                             Pages Author
read_time
##
## 1 Empath: A Complete Guide for Developing Your Gif…   104 Dyer, Judy
0
## 2 The Long War (The Long Earth, #2)                   419 Pratchett, …
8
## 3 The Robber Bride                                    528 Atwood, Mar…
9
## 4 When We Were Orphans                                320 Ishiguro, K…
0
```