```
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
```

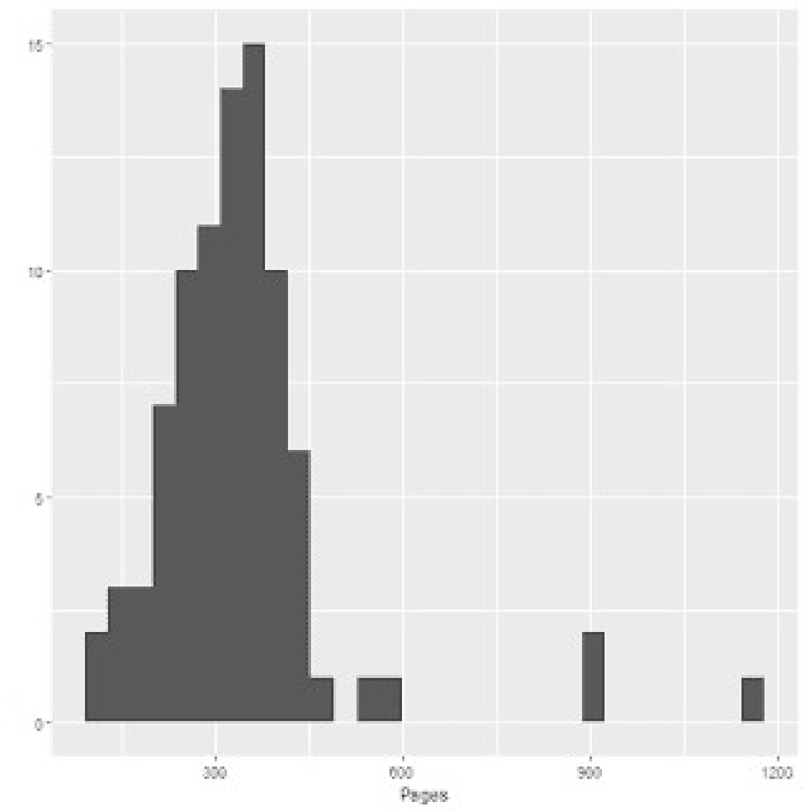
```
## -- Conflicts ----------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
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()
## )
```

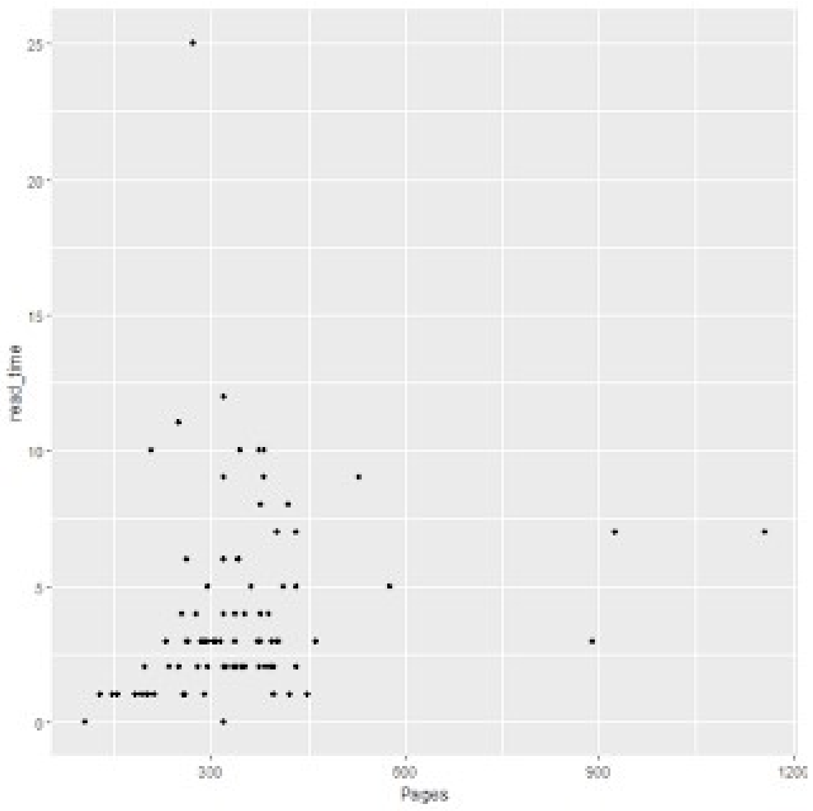
```
qplot(Pages, data = reads2019)
```

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



If I give it 2 continuous variables, it generates a scatterplot.

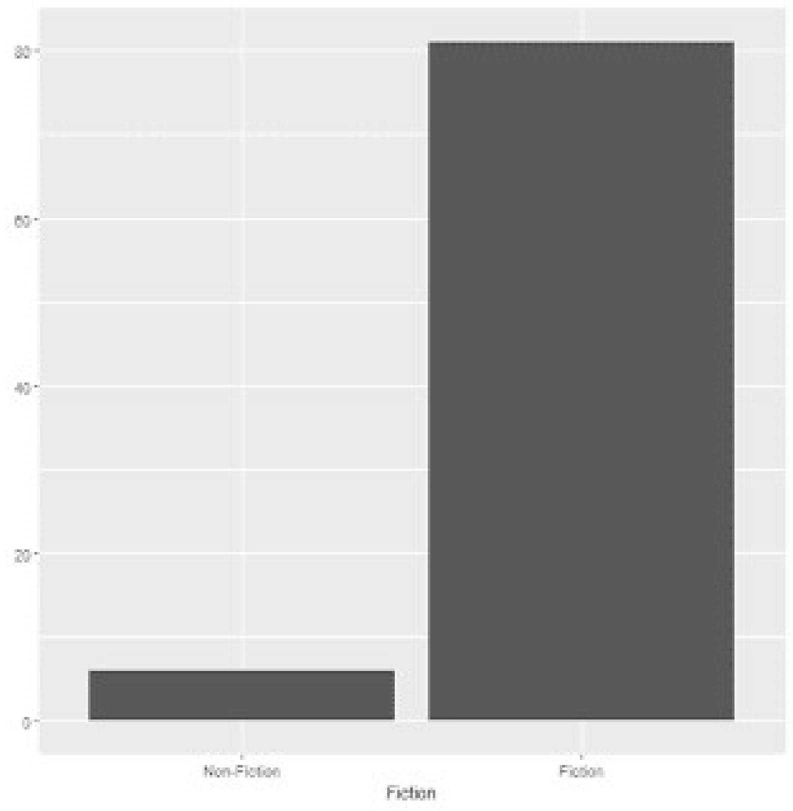
```
qplot(Pages, read_time, data = reads2019)
```



If I give it a factor, it generates a bar chart.

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

```
qplot(Fiction, data = reads2019)
```



And so on. Now, all throughout this series, and in many other stats and R posts, I've been using ggplot, instead of qplot. And I rarely, if ever, use qplot these days, even for quick data visualizations. Unlike qplot, ggplot requires you to specify the exact type of plot you want to make, using a geom_ in your code. And if you request a geom type that can't be made with the type of data you have, you'll get an error. Making a ggplot requires you to know what type of data you have and how you'd like it to be visualized. Despite (or perhaps because of) this, I strongly prefer ggplot and would encourage you to use it as well.

There are nearly endless ways to customize a ggplot – transforming scales, adding color schemes, layering data, changing fonts – that allow you make a fancy, schmancy, publication-ready plot. If you plan on publishing or presenting your work, you really want to use ggplots instead of qplots. qplots are like the first draft of your manuscript – no one sees it but you. But, in this case, it's a first draft that you can skip right over; just go straight to the good-looking plot.
Second, you notice that I didn't use the main pipe for my qplots above. That's because if you try it, you get an error.

```
reads2019 %>%
  qplot(Pages, read_time)
```

```
## Error in FUN(X[[i]], ...): object 'read_time' not found
```

The exposition pipe works, though.

```
library(magrittr)
```

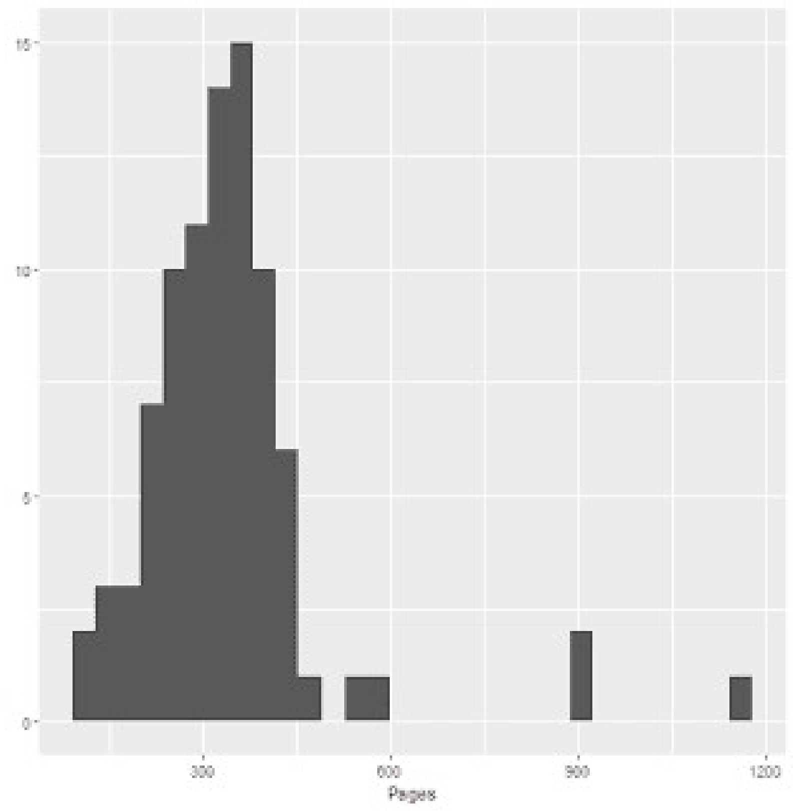
```
##
## Attaching package: 'magrittr'
```

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

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

```
reads2019 %$%
  qplot(Pages)
```

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



But the bad thing about the main pipe not working is that means you can’t combine any tidyverse verbs, like filters, with your plot code. You’d have to create a separate version of your data then use it with the qplot. For really large datasets, you might not have a lot of memory to spare. ggplot, on the other hand, works wonderfully with the main pipe.

ggplot really is the more powerful, prettier, endlessly customizable approach to plotting data. It can be challenging to get the hang of it at first, and it’s okay to ease your way into it. There are many great resources out there for learning the ggplot2 package, and specifically the ggplot function of that package. …