One of the easiest ways to make a beautiful ggplot is by using a theme. ggplot2 comes with a variety of pre-existing themes. I'll use the genre statistics summary table I created in yesterday's post, and create the same chart with different themes.

```
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
```

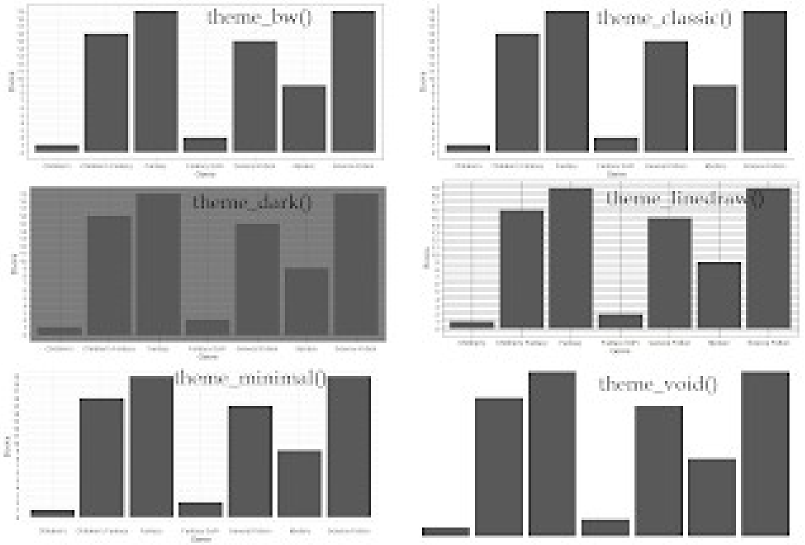
```
## -- Conflicts ----------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
reads2019 <- read_csv("~/Downloads/Blogging A to Z/SaraReads2019_allrated.csv",
                      col_names = TRUE)
```

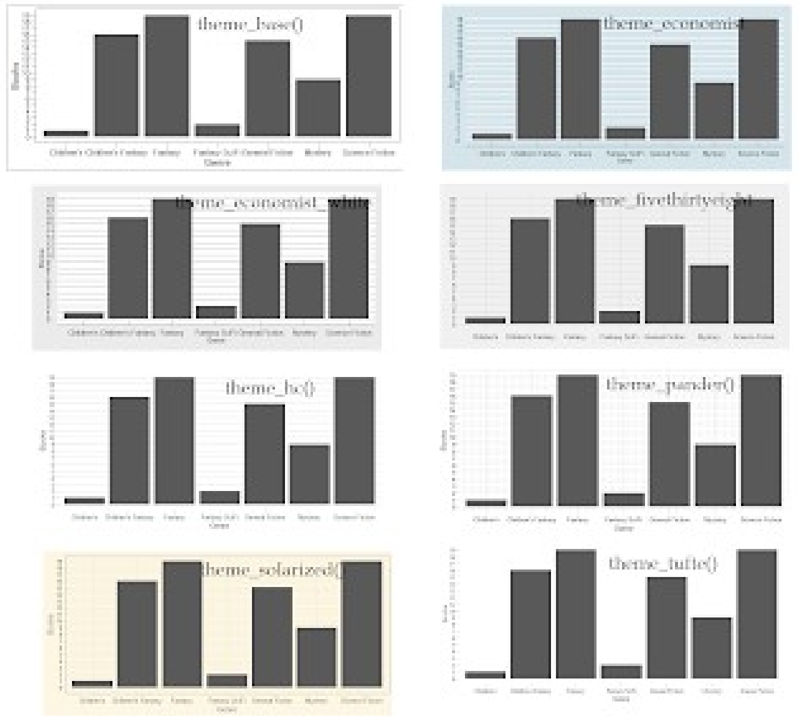
```
## Parsed with column specification:
## cols(
##   Title = col_character(),
##   Pages = col_double(),
##   date_started = col_character(),
##   date_read = col_character(),
##   Book.ID = col_double(),
##   Author = col_character(),
##   AdditionalAuthors = col_character(),
##   AverageRating = 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()
## )
```

```
genrestats <- reads2019 %>%
  filter(Fiction == 1) %>%
  arrange(OriginalPublicationYear) %>%
  group_by(Childrens, Fantasy, SciFi, Mystery) %>%
  summarise(Books = n(),
            WomenAuthors = sum(Gender),
            AvgLength = mean(Pages),
            AvgRating = mean(MyRating))


genrestats <- genrestats %>%
  bind_cols(Genre = c("General Fiction",
                  "Mystery",
                  "Science Fiction",
                  "Fantasy",
                   "Fantasy SciFi",
                   "Children's Fiction",
                   "Children's Fantasy")) %>%
  ungroup() %>%
  select(Genre, everything(), -Childrens, -Fantasy, -SciFi, -Mystery)

genre <- genrestats %>%
  ggplot(aes(Genre, Books)) +
  geom_col() +
  scale_y_continuous(breaks = seq(0,20,1))
```

Since I’ve created a new object for my figure, I can add a theme by typing genre + [theme]. Here’s a handful of the ggplot2 themes.

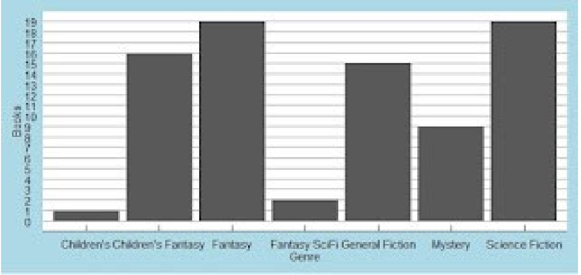


You can also get more themes with additional packages. My new favorite is ggthemes. I’ve been loving their Economist themes (particularly economist_white), which I’ve been using for most of the plots I create at work. Here are some of my favorites.



You can also customize different elements of the plot with theme(). For instance, theme(plot.title = element_text(hjust = 0.5)) centers your plot title. theme(legend.position = “none”) removes the legend. You could do both of these at once within the same theme() by separating them with commas. This is a great way to tweak tiny elements of your plot, or if you want to create your own custom theme.

```
library(ggthemes)

## Warning: package 'ggthemes' was built under R version 3.6.3

genre +
  theme_economist_white() +
  theme(plot.background = element_rect(fill = "lightblue"))
```



These themes also have color schemes you can add to your plot. We'll talk about that soon!