Eric_Hirsch_607_TidyVerse

Eric Hirsch

2021-02-17

ggplot2 - Creating Elegant and Useful Graphs in R

`ggplot2`, a package in the core tidyverse, is a system for easily and efficiently creating graphics in R. It is based on the "Grammar of Graphics", the notion that all graphs can be built from the same components: a data set, a coordinate system, and visual marks that represent data points. You specify these details to ggplot2 and the package creates the graph.

The possibilities with ggplot2 are vast. You can learn more at https://ggplot2.tidyverse.org/. I will describe here some of the basic core capabilities.

Loading the library and our sample data

We start by loading the tidyverse library. If tidyverse is not installed you must first install the tidyverse package (install.packages("tidyverse"))

```
library(tidyverse)
#> -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
#> v ggplot2 3.3.2     v purrr   0.3.4
#> v tibble  3.0.4     v dplyr   1.0.2
#> v tidyr   1.1.2     v stringr 1.4.0
#> v readr   1.4.0     v forcats 0.5.0
#> -- Conflicts ------------------------------------------ tidyverse_conflicts() --
#> x dplyr::filter() masks stats::filter()
#> x dplyr::lag()    masks stats::lag()
```

Next we load the data that will be used for our examples. This data is from the article, "Marriage Isn't Dead - Yet", published on fivethirtyeight.com. The data represents the share of the population, aged 25 to 34, that has never been married, broken down by year and by other factors - we will be looking at level of education.

```
dfMarriage <- read.csv("https://raw.githubusercontent.com/fivethirtyeight/data/master//marriage/both_se
dfMarriage_Subset <- subset(dfMarriage, select = c("year", "all_2534", "HS_2534", "SC_2534", "BAo_2534"
dfMarriage_Subset <- rename(dfMarriage_Subset, c(all_individuals =  "all_2534", High_School_Or_Less = "

head(dfMarriage_Subset)
#>   year all_individuals High_School_Or_Less Some_College Bachelors_NoGradDegree
#> 1 1960       0.1233145           0.1095332    0.1522818              0.2389952
#> 2 1970       0.1269715           0.1094000    0.1495096              0.2187031
#> 3 1980       0.1991767           0.1617313    0.2236916              0.2881646
#> 4 1990       0.2968306           0.2777491    0.2780912              0.3656655
```

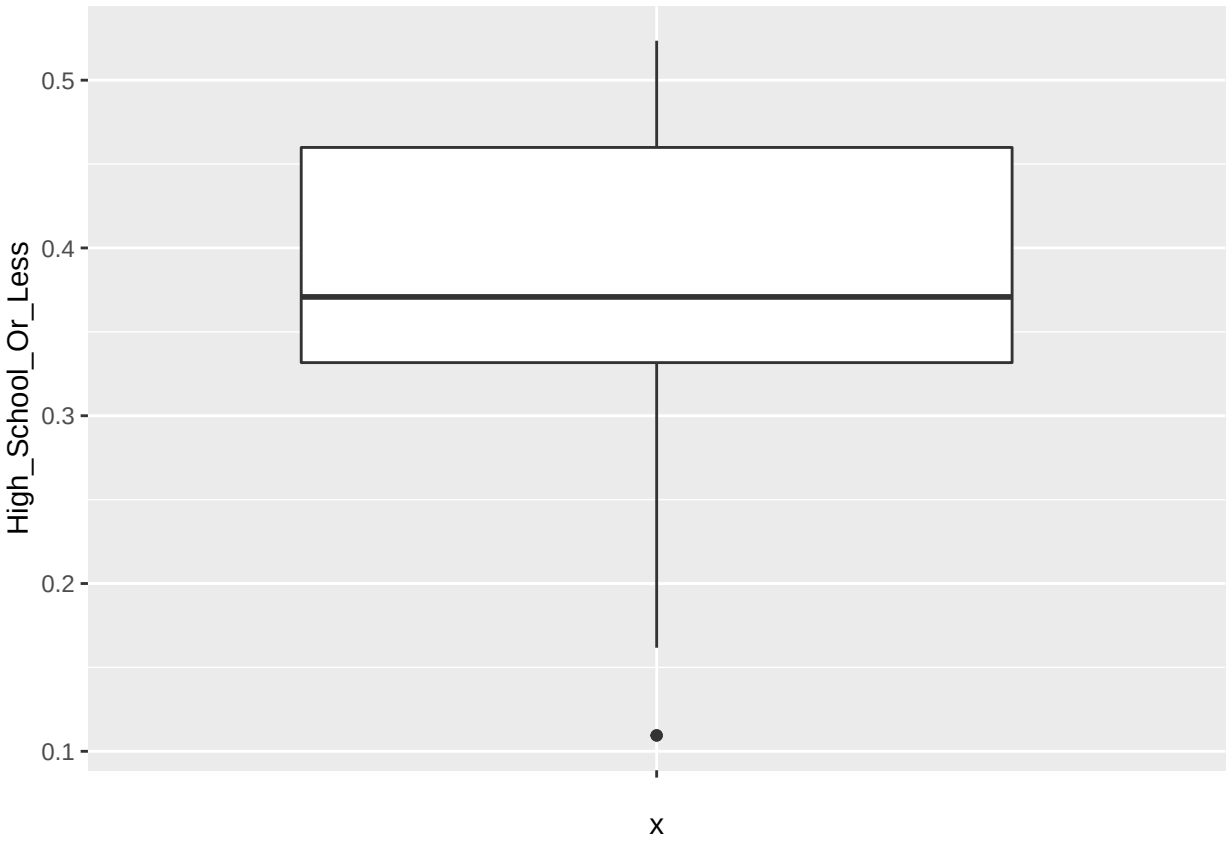
```
#> 5 2000       0.3450087           0.3316545    0.3249205           0.3939579
#> 6 2001       0.3527767           0.3446069    0.3341101           0.3925148
#>   Graduate_Degree
#> 1              NA
#> 2              NA
#> 3              NA
#> 4       0.3474505
#> 5       0.3691740
#> 6       0.3590304
```

Using the ggplot2 library

ggplot2 can create a vast array of graphs with many features. We will focus here on a few: creating different types of graphs, and adding simple features like titles and labels.

We begin with a boxplot of the share of never-married adults in the population with a high school education or less, for all the years in the dataset:

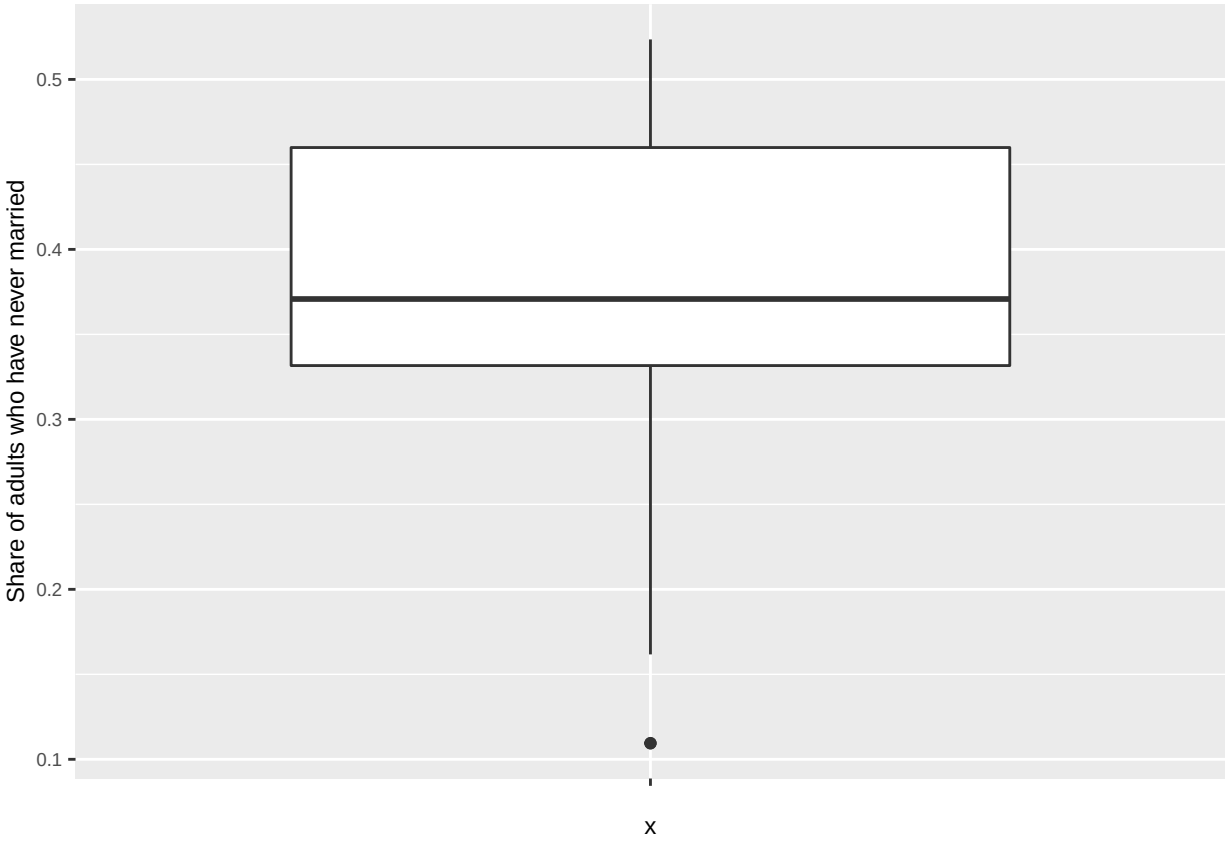
```
ggplot(data = dfMarriage_Subset, aes(x ="", y=High_School_Or_Less)) +
 geom_boxplot()
```



Note the grammar of graphics at work - we supply a dataset (*dfMarriage_subset*), a coordinate system, or "aesthetic" (*aes(x = "", y=High_School_Or_Less))*) and a type of graph (*geom_boxplot()*) - and ggplot2 does the rest.

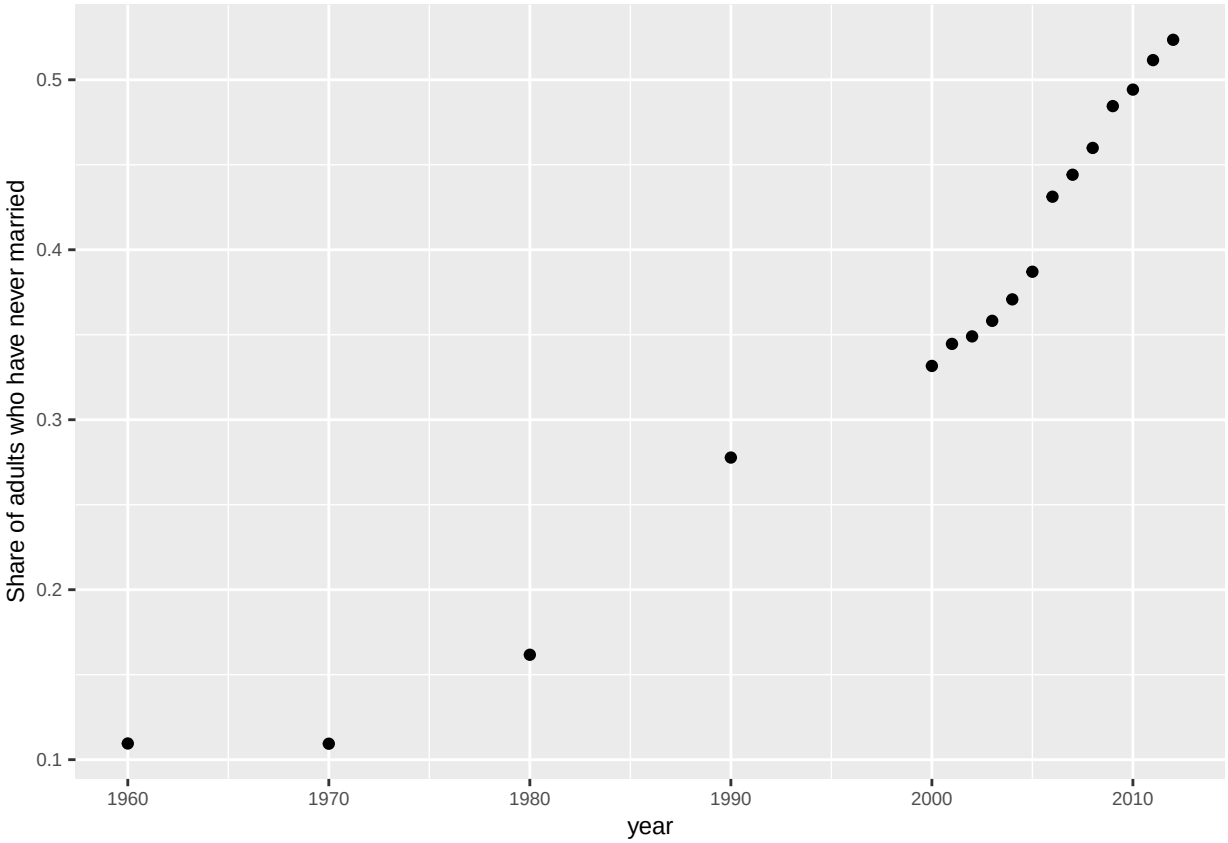
If we change or add to any of the parameters for this plot, we get a new graph. For example, we can improve the y axis label by specifying new text with *ylab()* and a new font size with *theme(text = element_text(size = 9))*.

```
ggplot(data = dfMarriage_Subset, aes(x ="", y=High_School_Or_Less)) +
 geom_boxplot() +
  ylab("Share of adults who have never married") +
  theme(text = element_text(size = 9))
```



What about a scatterplot of the share of 'never-married adults with a high school education or less' against 'year'? True to the grammar of graphics, we can use the same plot with two adjustments - change the x axis to 'year' and the plot type to 'geom_point.' The plot will retain our y axis label:

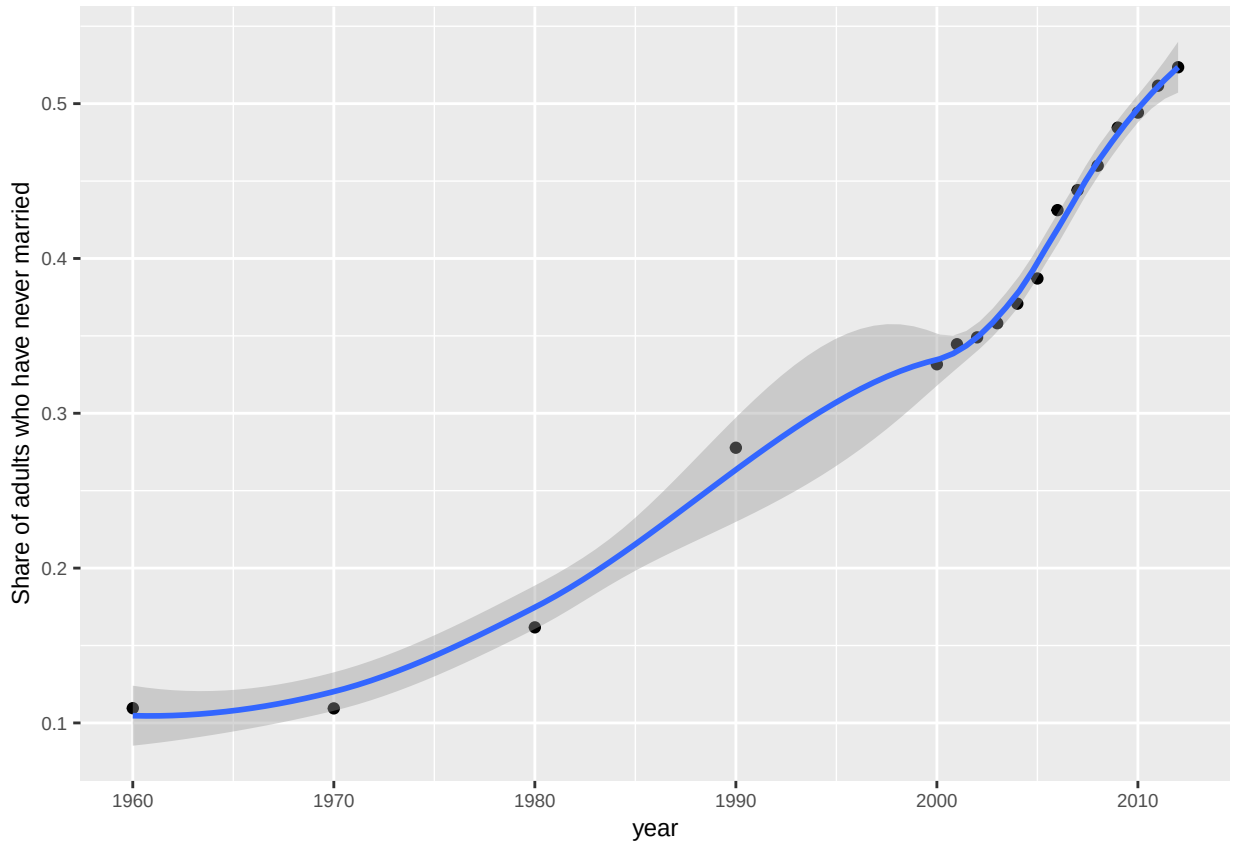
```
ggplot(data = dfMarriage_Subset, aes(x=year, y=High_School_Or_Less)) +
 geom_point() +
  ylab("Share of adults who have never married") +
  theme(text = element_text(size = 9))
```



We can see that among adults between 24 and 34 with a high school education or less, the share of those who have never been married has increased steadily over time.

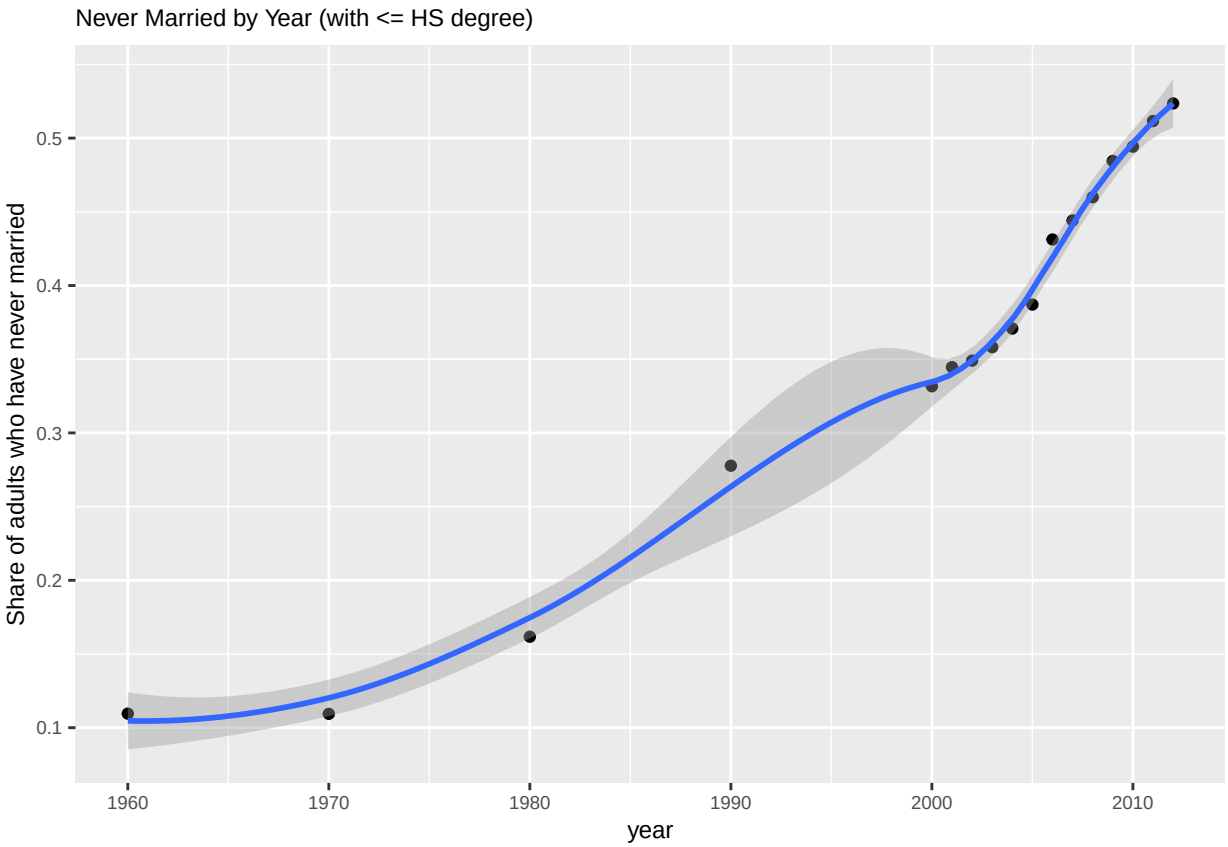
We can continue to add features. Let's smooth out the plot by adding *geom_smooth ()*.

```
ggplot(data = dfMarriage_Subset, aes(x = year, y=High_School_Or_Less)) +
  geom_point() +
  ylab("Share of adults who have never married") +
  theme(text = element_text(size = 9))+
  geom_smooth ()
#> `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Now let's finish our graph by including a general title, and by setting the font size for the new title so it fits on the graph. To do this we add *ggtitle(title)* and *theme(plot.title = element_text(size = 9))*

```
ggplot(data = dfMarriage_Subset, aes(x = year, y=High_School_Or_Less)) +
  geom_point() +
  ylab("Share of adults who have never married") +
  theme(text = element_text(size = 9))+
  geom_smooth () +
  ggtitle("Never Married by Year (with <= HS degree)") +
  theme(text = element_text(size = 9), plot.title = element_text(size = 9))
#> `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



There is so much more we can do! But this introduction should get you started.