Visualizing Descriptives

Matt Lawrence

September 23, 2019

Setting Up

This week we will continue using the 2000-2014 waves of the General Social Survey. We are still focused on the `agekdbrn` variable but will start looking at how its distribution varies by education and race/ethnicity. Start by loading tidyverse and the data.

Variance and Standard Deviation

To find the variance, use `var()`. To find the standard deviation, use `sd()`.

```
var(gss_week3$agekdbrn)
```

```
## [1] 31.41344
```

```
sd(gss_week3$agekdbrn)
```

```
## [1] 5.604769
```

Would you expect more or less variation in the education distribution?

REPLACE THIS LINE WITH YOUR CODE

```
var(gss_week3$educ)
```

```
## [1] 6.906906
```

```
sd(gss_week3$educ)
```

```
## [1] 2.628099
```

Introducing ggplot2

Let's look at some visualizations of the distribution of age at first birth. There are two main types of visualizations you can make with R: base graphics and ggplot2 graphics. We are going to focus on ggplot2, one of the packages (like dplyr) that loads with the tidyverse.

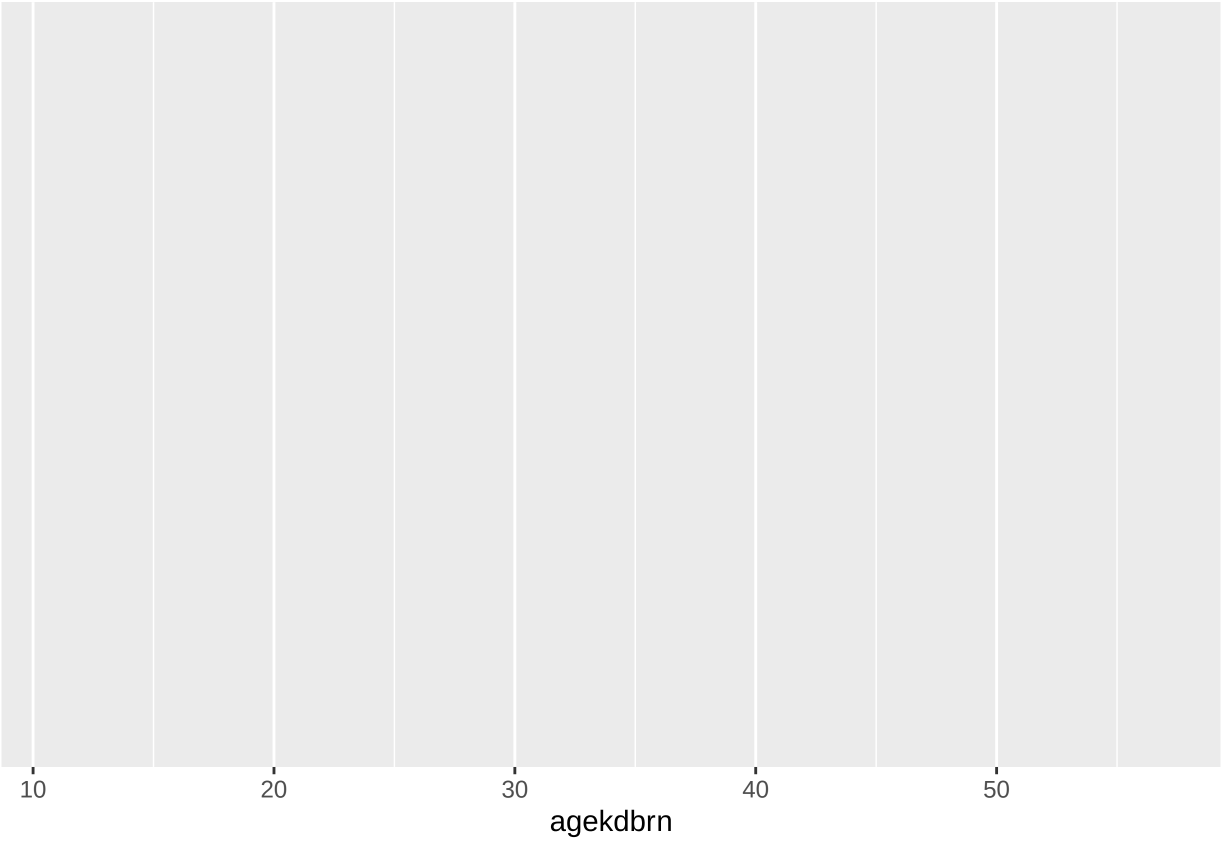
The "grammar of graphics" that ggplot is built on creates graphics layer by layer, mapping aesthetics and options on top of other layers. Let's see how this works in a very basic density plot.

The first step is to name our figure. In this example, we'll save it as an object called `agekdbrn_plot`. The right side of the following chunk says that we will be using the ggplot functions with the `gss_week3` data frame, and the `aes` parentheses is where we say what we want on the x and y axes. For this initial plot, we will only have a variable on the x axis.

```
agekdbrn_plot <- ggplot(gss_week3, aes(x = agekdbrn))
```

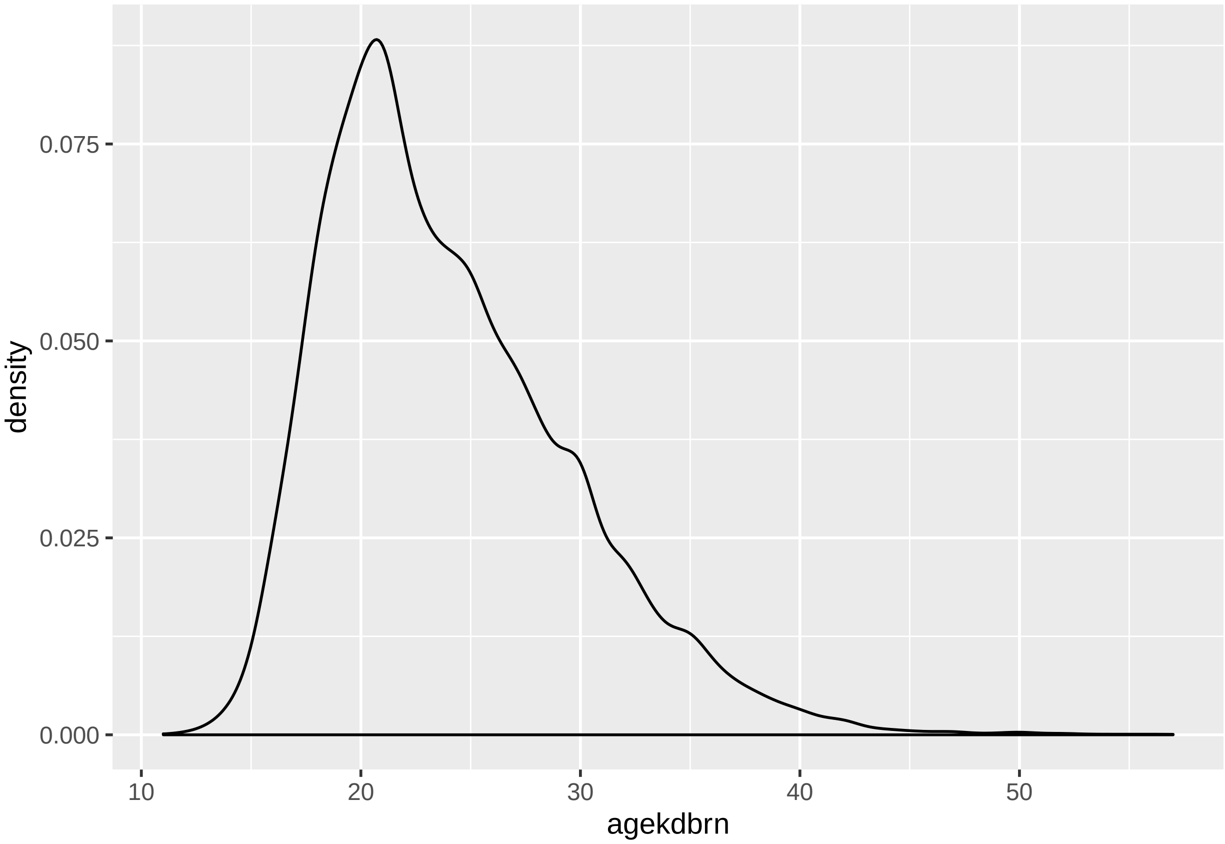
To get a sense of how plots are built layer by layer, let's take a look at what is now connected to our object:

```
agekdbrn_plot
```



So R knows the range of our variable and has used that info to set the backdrop for our plot. The next step is to add a layer on top of this. We use our existing object name and add the geometic feature we want. In this example, we will look at a density plot:

```
agekdbrn_plot + geom_density()
```

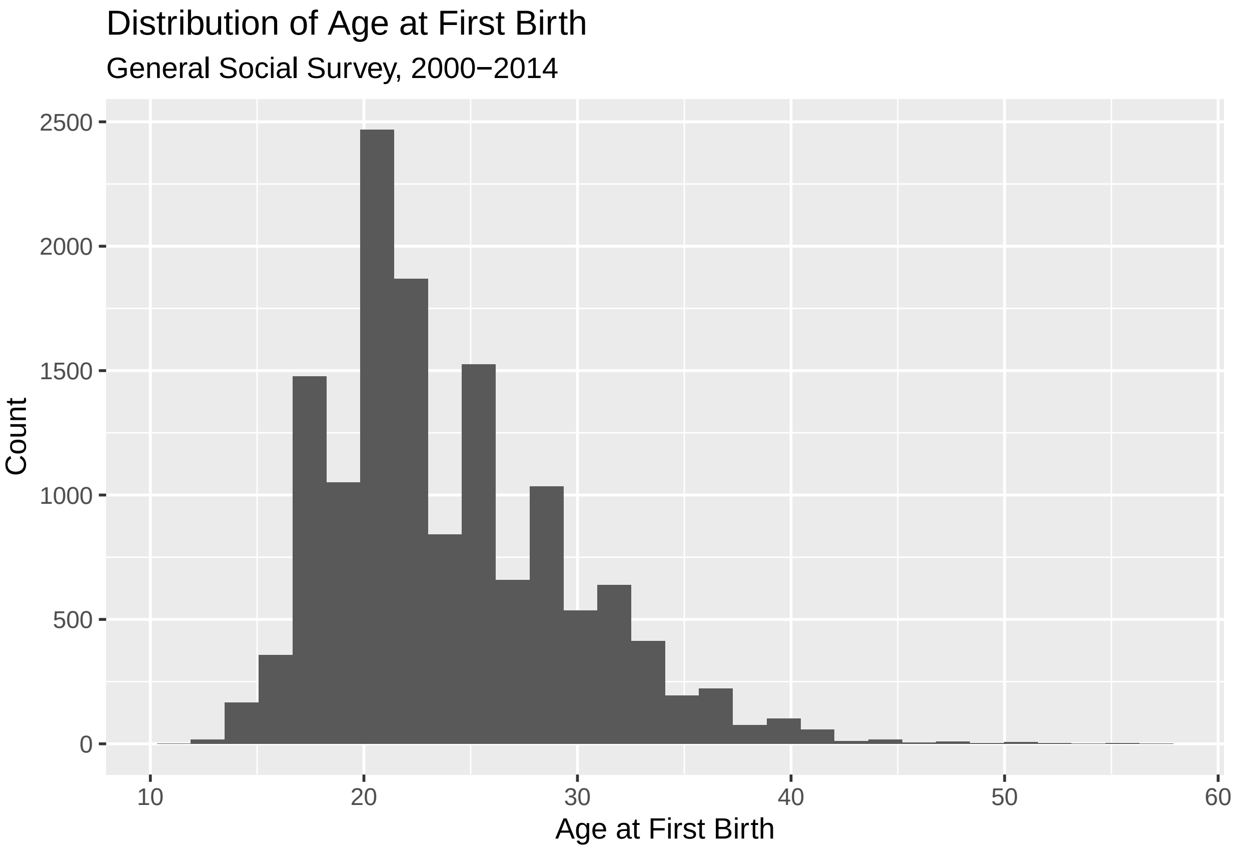


What is this distribution telling us? How would you define the shape and the skew?

There are many ways to improve this plot. We'll start simply by changing the axis labels and adding a title. If you are making all of these changes, the easiest thing to do is to use the `labs()` function:

```
agekdbrn_plot + geom_histogram() +
  labs(x = "Age at First Birth", y = "Count",
        title = "Distribution of Age at First Birth",
        subtitle = "General Social Survey, 2000-2014")
```

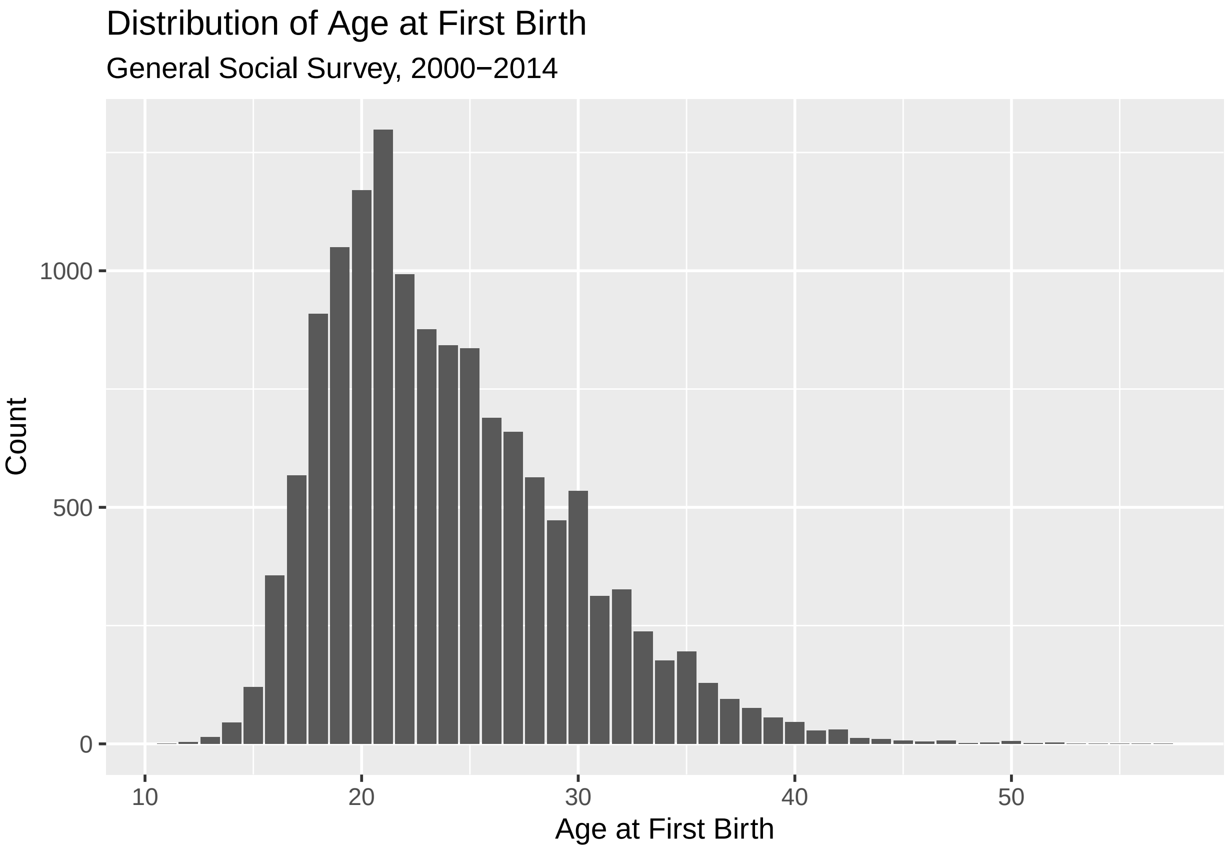
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Two other visualizations that are good for looking at distributions are bar plots and histograms. These use `geom_bar` and `geom_histogram` respectively. Try them out by editing the last code chunk we ran to replace `geom_density` with `geom_bar` or `geom_histogram`.

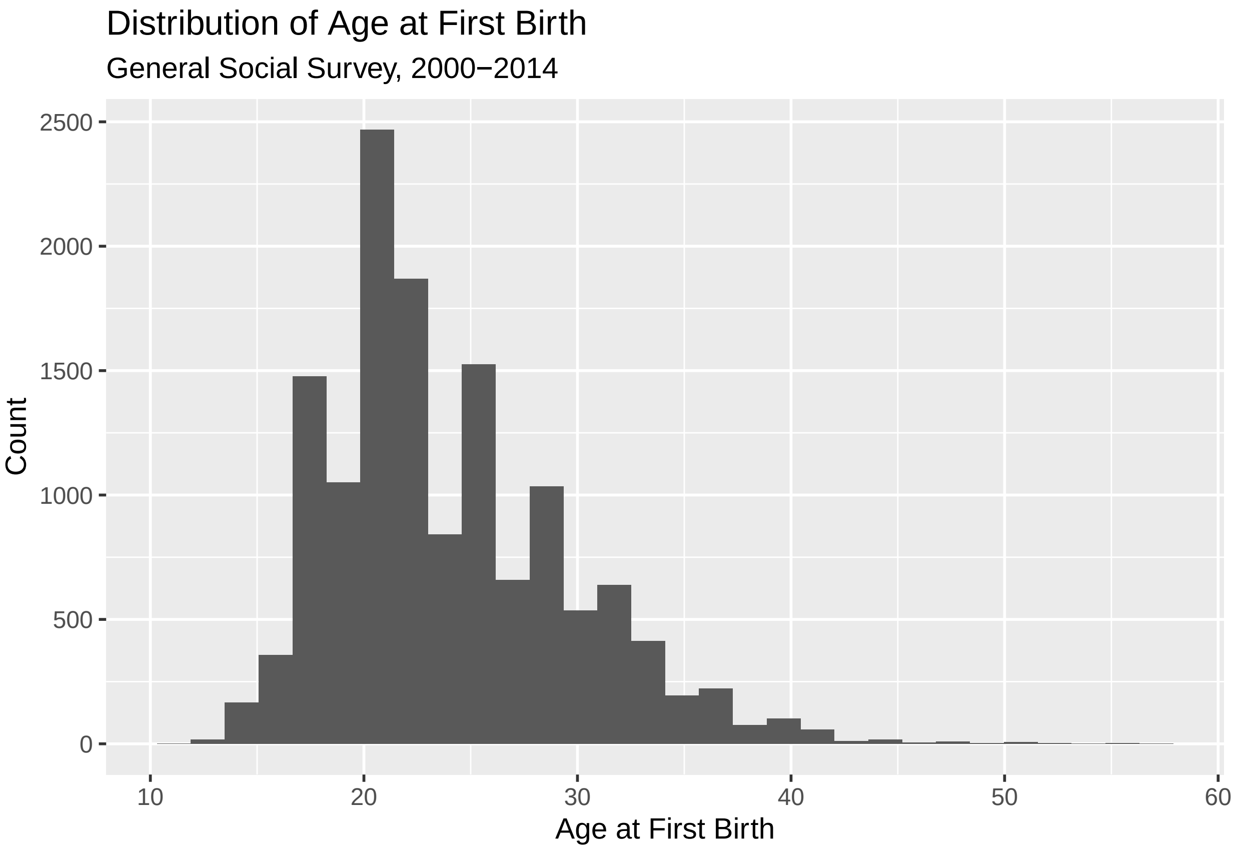
REPLACE THIS LINE WITH YOUR CODE

```
agekdbrn_plot + geom_bar() +
       labs (x = "Age at First Birth", y = "Count",
        title = "Distribution of Age at First Birth",
        subtitle = "General Social Survey, 2000-2014")
```



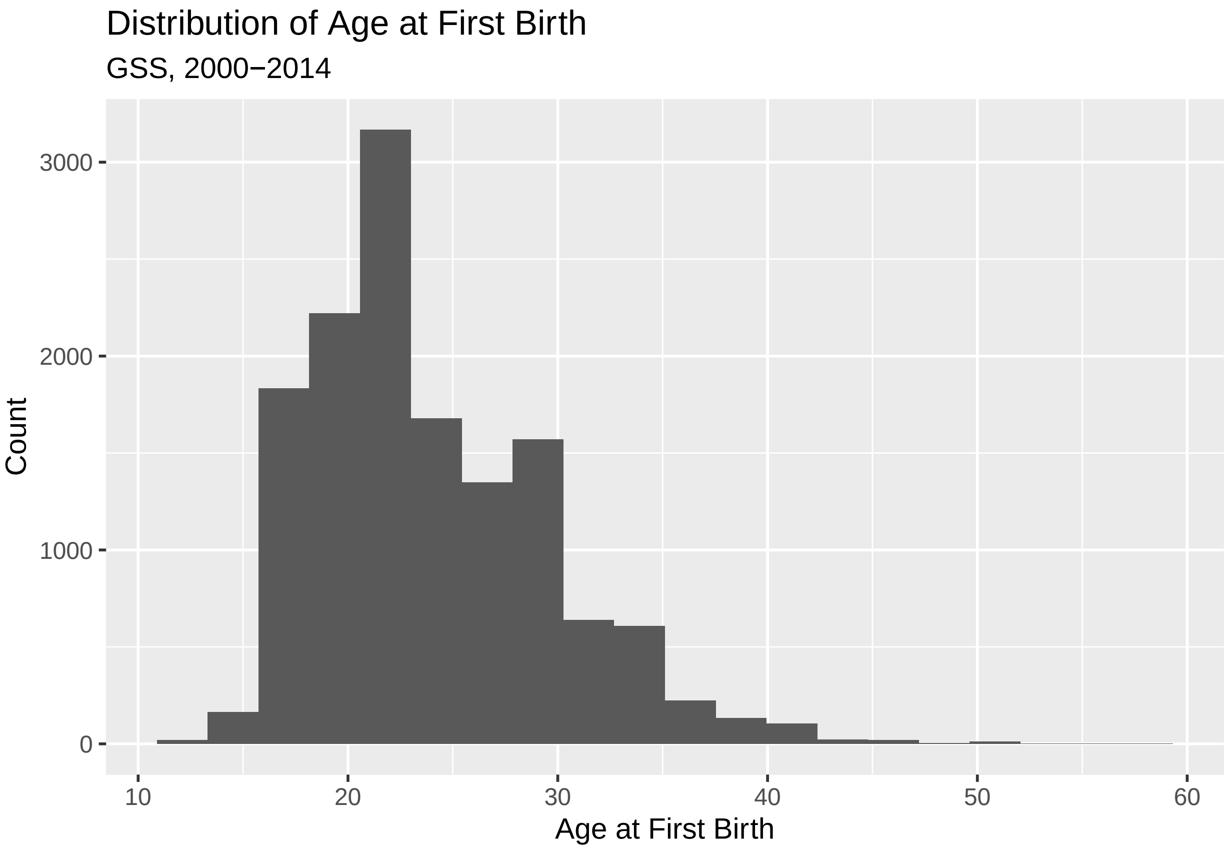
```
agekdbrn_plot + geom_histogram() +
       labs (x = "Age at First Birth", y = "Count",
        title = "Distribution of Age at First Birth",
        subtitle = "General Social Survey, 2000-2014")
```

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



The warning message about binwidth with the histogram is asking you to choose how many buckets you want the data grouped by. See what happens as you increase and decrease the number of bins:

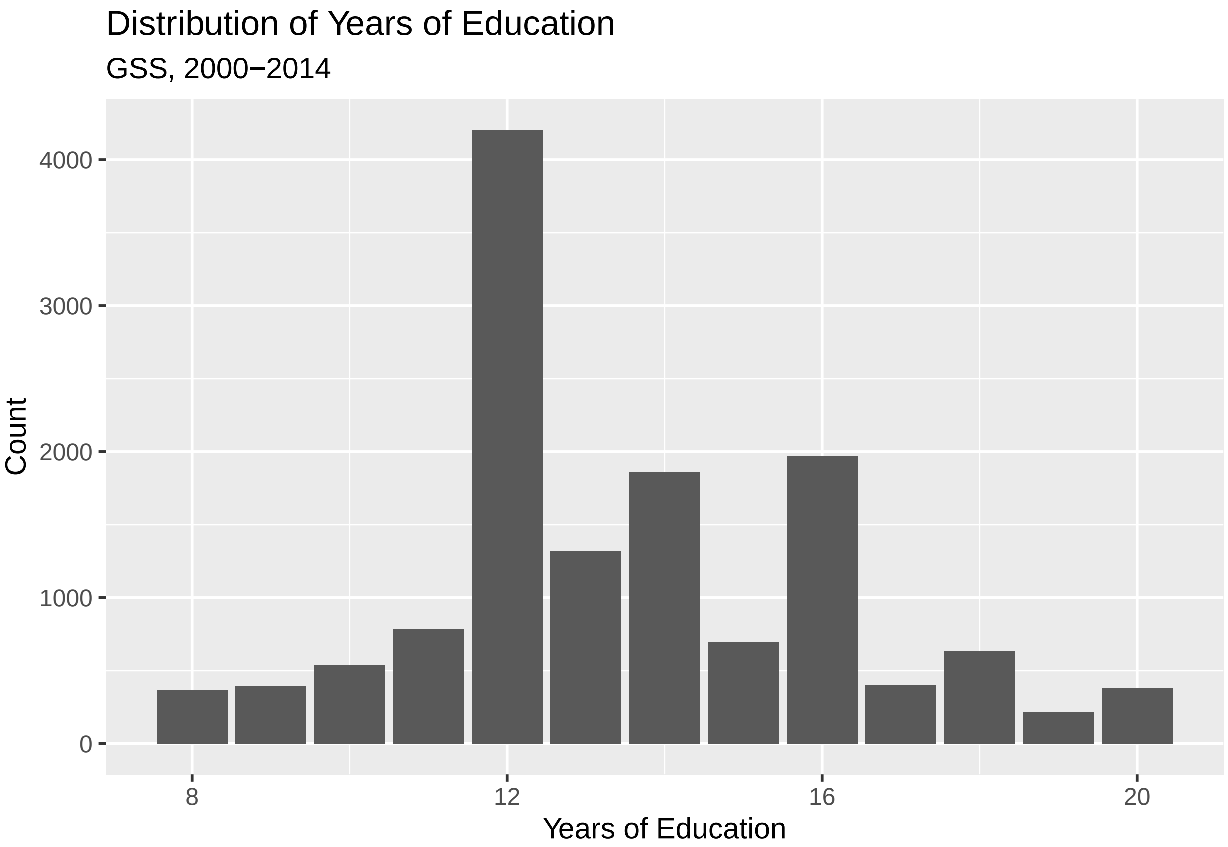
```
agekdbrn_plot + geom_histogram(bins = 20) +
  labs (x = "Age at First Birth", y = "Count",
        title = "Distribution of Age at First Birth",
        subtitle = "GSS, 2000-2014")
```



Try a bar plot showing the distribution of the `educ` variable.

REPLACE THIS LINE WITH YOUR CODE

```
educ_barplot <- ggplot(gss_week3, aes(x = educ))
educ_barplot + geom_bar() +
     labs(x = "Years of Education",
          y = "Count",
          title = "Distribution of Years of Education",
          subtitle = "GSS, 2000-2014")
```



Creating Categories

The plots above have visualized the distributions across all respondents. But what if we want to compare distributions across groups of respondents? Suppose we want to compare the distribution of age at first birth across education groups. We already know how to use indexing to get the values of one variable for observations that have specific values of another variable. That would be a good start for finding the average age at first birth for different ranges of the education distribution. But it would be more helpful if we could collapse the values of the education variable into categories.

Let's start with a binary variable distinguishing respondents with college degrees from respondents without college degrees. We can do this in dplyr using mutate() and the ifelse() function. In this example, we'll create a new variable called `college_degree` and we'll put it in the `gss_week3` data frame.

```r
gss_week3 <- gss_week3 %>%
     mutate(college_degree = ifelse(educ<16, 0, 1))
```

Read the mutate line as:

"Create a new variable called `college_degree`. If the value of the education variable is less than 16, the new variable's value should be 0. If the value of the education variable is anything else, the new variable's value should be 1."

Use this new variable to find the proportion of respondents with a college degree.

REPLACE THIS LINE WITH YOUR CODE

```
summary(gss_week3$college_degree)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.0000  0.0000  0.0000  0.2619  1.0000  1.0000
```

Binary variables are generally very helpful. For plotting, though, factor variables are better. We can use mutate() again to assert that we want `college_degree` to be a factor variable, and we will add the `labels = c()` option to label 0 “No” and 1 “Yes”.

```
gss_week3 <- gss_week3 %>%
     mutate(college_degree = factor(college_degree,
                                    labels = c("No", "Yes")))
```

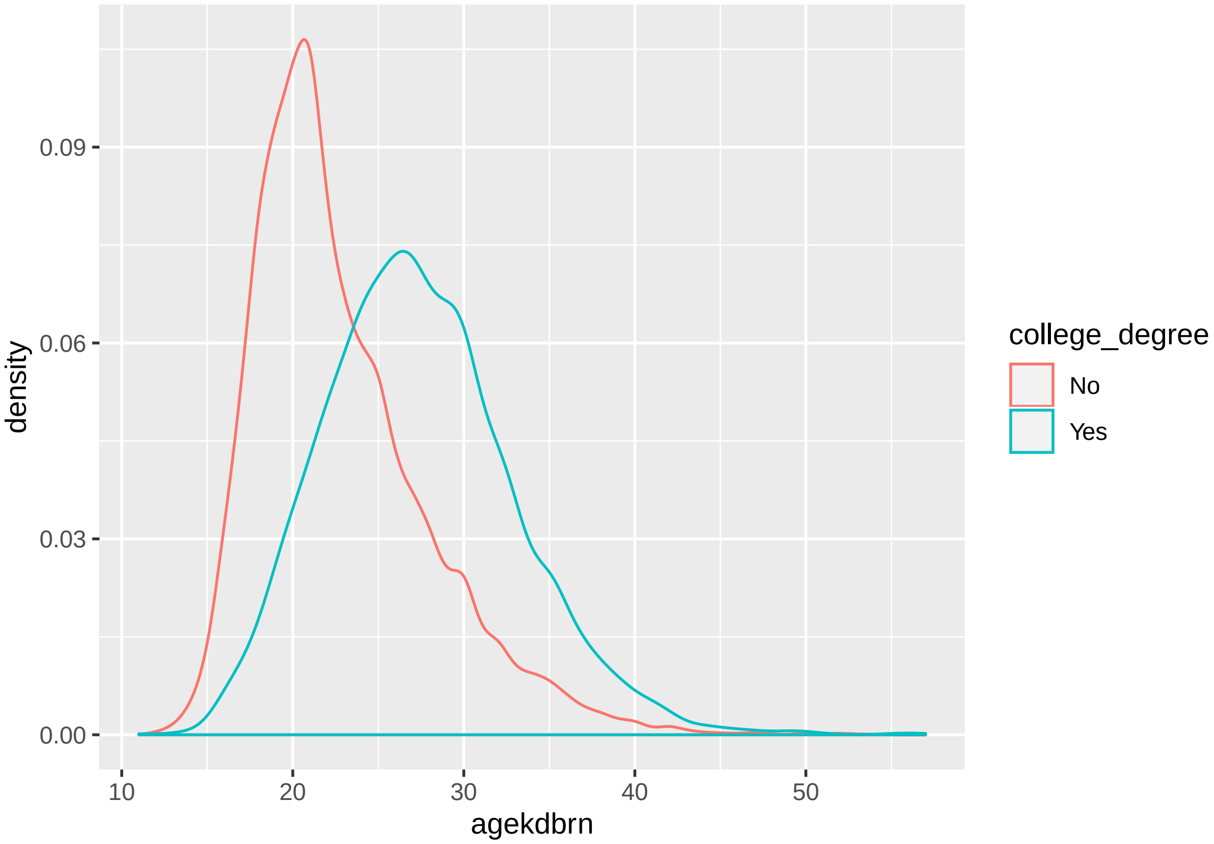
A proportion table should show the same proportions of respondents with college degrees that we saw above with the binary variable.

```
round(prop.table(table(gss_week3$college_degree)),3)
```

```
## 
##    No   Yes 
## 0.738 0.262
```

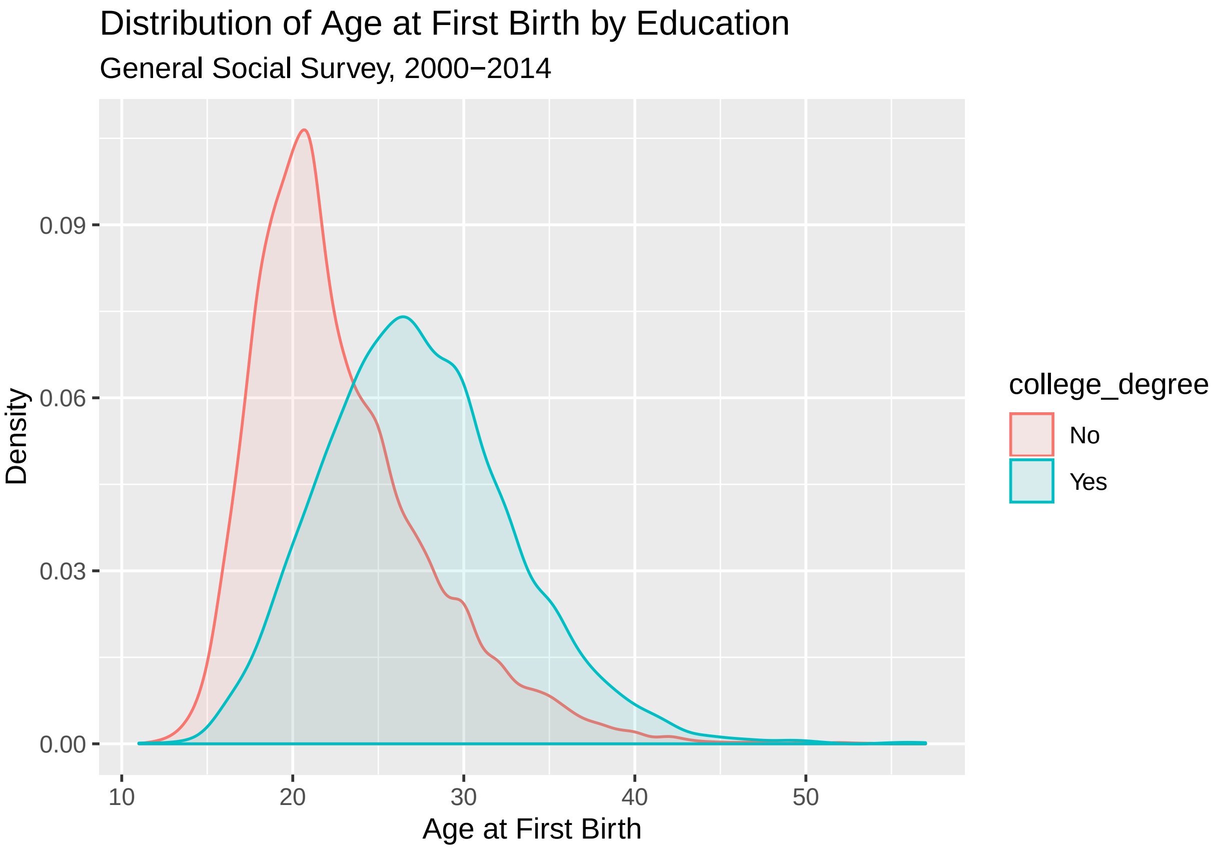
Now let’s make a density plot with two curves: one for respondents without college degrees and one for respondents with college degrees. We will add the `color =` option to the aesthetic mapping to assert that we want each curve outlined in a different color based on the two values of the `college_degree` variable.

```
agekdbrn_college_plot <- ggplot(gss_week3,
                                aes(x = agekdbrn, color = college_degree))
agekdbrn_college_plot + geom_density()
```



How could we improve this plot? Let's color in the areas under the curves using the `fill =` option. Let's use `alpha =` to adjust the transparency of the overlapping density curves so they do not completely cover each other. And let's fix the labels.

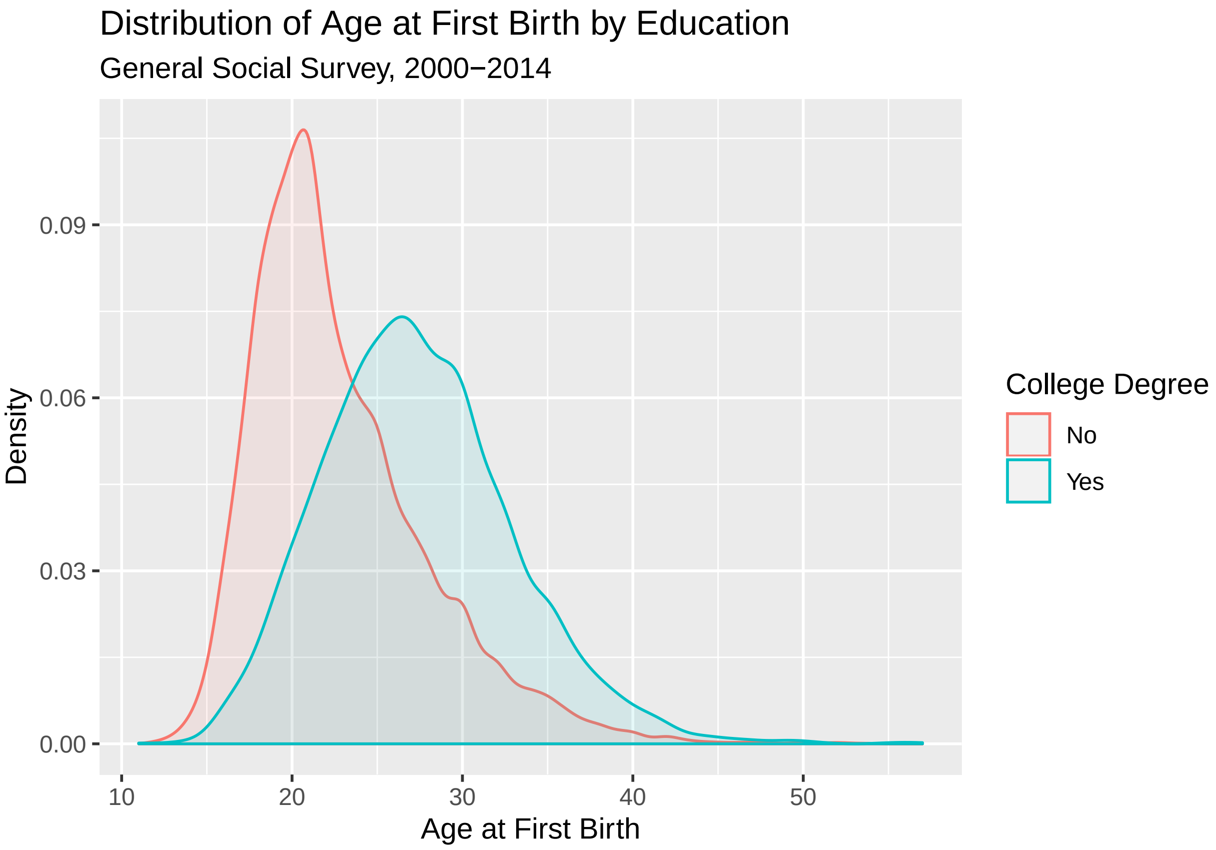
```
agekdbrn_college_plot <- ggplot(gss_week3,
                                aes(x = agekdbrn,
                                color = college_degree,
                                fill = college_degree))
agekdbrn_college_plot + geom_density(alpha = .1) +
     labs(x = "Age at First Birth",
          y = "Density",
          title = "Distribution of Age at First Birth by Education",
          subtitle = "General Social Survey, 2000-2014")
```



Stop Here On Tuesday

Change legend?

```
agekdbrn_college_plot <- ggplot(gss_week3,
                                aes(x = agekdbrn,
                                color = college_degree,
                                fill = college_degree))
agekdbrn_college_plot + geom_density(alpha = 0.1) +
     labs(x = "Age at First Birth",
          y = "Density",
          title = "Distribution of Age at First Birth by Education",
          subtitle = "General Social Survey, 2000-2014") +
     scale_color_discrete(name = "College Degree") +
     scale_fill_discrete(guide = FALSE)
```

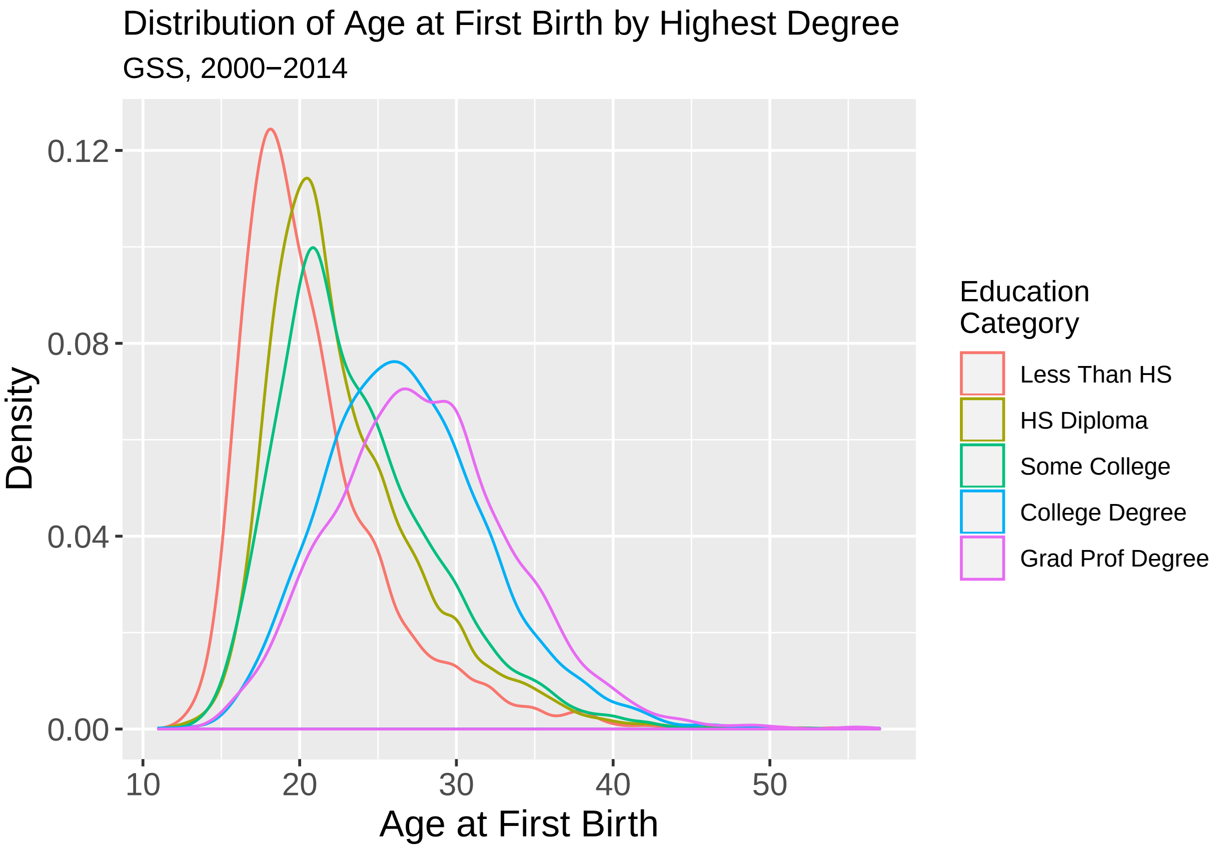


Preview of next class with education categories. . .

```
gss_week3 <- gss_week3 %>%
     mutate(edcat = ifelse(educ < 12, "Less Than HS",
                           ifelse(educ == 12, "HS Diploma",
                                  ifelse(educ %in% 13:15, "Some College",
                                         ifelse(educ == 16, "College Degree",
                                                "Grad Prof Degree")))))

gss_week3 <- gss_week3 %>%
     mutate(edcat = factor(edcat,
          levels = c("Less Than HS",
                     "HS Diploma",
                     "Some College",
                     "College Degree",
                     "Grad Prof Degree")))

agekdbrn_edcat_plot <- ggplot(gss_week3, aes(x = agekdbrn, color = edcat))
agekdbrn_edcat_plot + geom_density() +
  labs(x = "Age at First Birth",
       y = "Density",
       title = "Distribution of Age at First Birth by Highest Degree",
       subtitle = "GSS, 2000-2014") +
     scale_color_discrete(name = "Education\nCategory") +
     theme(axis.title=element_text(size=14),
           axis.text=element_text(size=12))
```


Distribution of Age at First Birth by Highest Degree
GSS, 2000−2014
Density
0.12
0.08
0.04
0.00
10
20
30
40
50
Age at First Birth
Education Category
Less Than HS
HS Diploma
Some College
College Degree
Grad Prof Degree