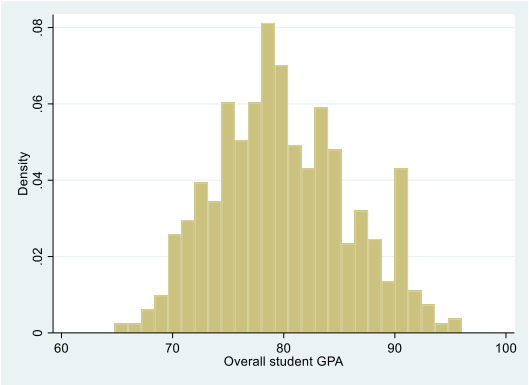
Visualizing data in Stata – Section 2.2
Najib Mozahem

Let us start by looking at the variable *gpa*. As we know by now, this variable measures the students' overall GPA so far. What would be very useful for us is to know the distribution of the grades. Are most of the grades above 85% for example? Or are they mostly less than 75%? Does our sample of students contain a large variety of GPAs or do most of the students have more or less similar values for the variables? When we have a continuous variable and we want to visualize its distribution, we need to use the command **histogram**:

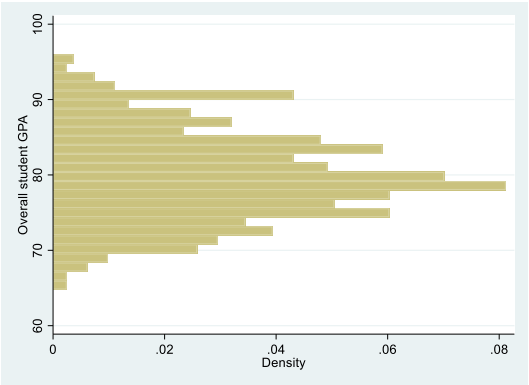
histogram gpa



As you can see, the command takes on only one variable, which in our case is *gpa*. As you can see, our sample includes students with a wide range of GPA values. There are some students with a GPA that is above 90 and some with a GPA that is less than 70. However, we note that most of the values tend to be concentrated near the middle, somewhere in the 75 – 85 range. This is actually what we expect when we look at grades in a university. Most students do average, with a minority of them either doing extremely well or extremely bad.

In some cases you might prefer to draw the histogram horizontally:

histogram gpa, horizontal

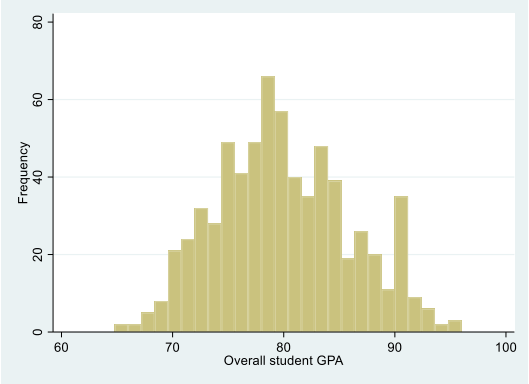


As you can see, Stata options make things extremely easy for us.

If you look at the y-axis (the vertical axis) you will notice that it is labeled as "Density". The larger the density of the bar, the larger the number of students that fall within its boundaries. This notion

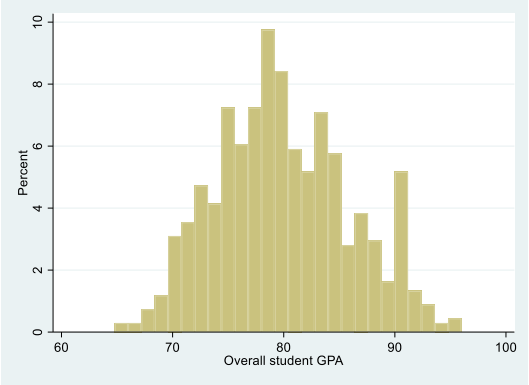
of "density" isn't very intuitive. It is calculated in such a way as to make the sum of the areas of all the bars equal to one. A better option would be to display frequency on the y-axis:

histogram gpa, frequency



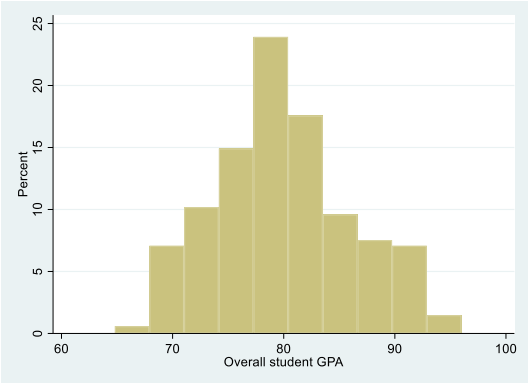
In this case, the height of each bar represents the number of students included within the boundaries of the bar. Another option might be to see the percentages instead of the frequencies:

histogram gpa, percent



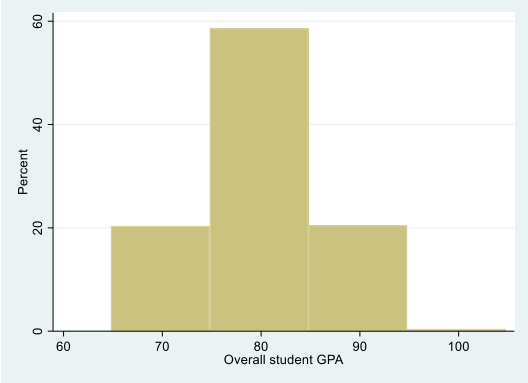
We can also control the number of bars that are shown. Perhaps you think that Stata has produced too many categories. You can tell Stata to show whatever number of bars, or bins, you want:

histogram gpa, bin(10) percent



We notice now that the number of bars is 10. When we specify the number of bars, Stata calculates the range, or width, of each bar. The largest GPA in our dataset is 96 and the smallest is 64.79. By telling Stata that we want to have 10 bars, Stata calculates that the width of each should be around (96-64.79)/10 = 3.12. What if we wanted to specify the width of each bar instead of specifying the number of bins? This is also possible:

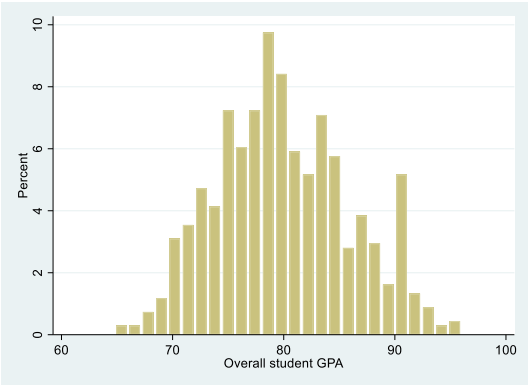
histogram gpa, width(10) percent



In this command we told Stata that we want the width of each bin to be 10. Looking at the graph we see that the three large bins cover 10 points (from 75 to 85 for the middle one for example). We also notice that there is a very short bin at the very end. This is because the largest GPA is 96 and it lies outside the range of the bin that is centered around 90.

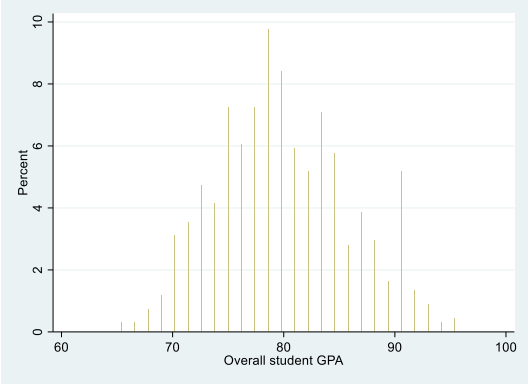
We can also customize not just the number and the width of bars, but also the way that they look. The following command controls the size of the space between two consecutive bars:

histogram gpa, gap(20) percent



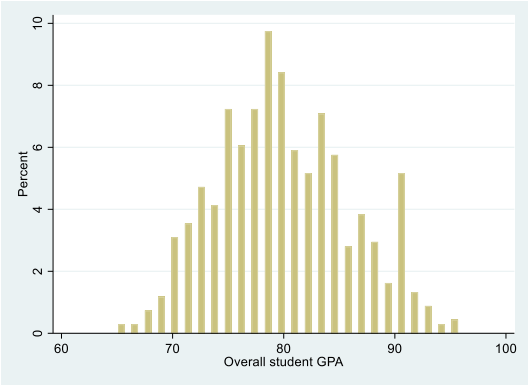
The **gap()** option helps us specify the size of the gap between two consecutive bars. The value that we specify must be between 0 and 100. We can try specifying a larger value:

histogram gpa, gap(99) percent



Instead of specifying the width of the gap between two consecutive bars, we can specify the width of each bar:

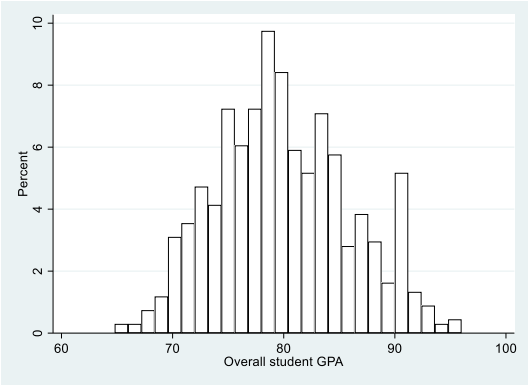
histogram gpa, barwidth(0.6) percent



Notice that we used the **barwidth()** option and not the **width()** option. This is a very important point because **barwidth()** changes how the bars are drawn, but the **width()** option changes how they are calculated because it specifies the range of each bar.

Stata also allows us to control the colors used in drawing the graphs. To do this we can use the three options: **fcolor()**, **lcolor()**, and **lwidth()**:

histogram gpa, percent fcolor(white) lcolor(black) lwidth(thin)

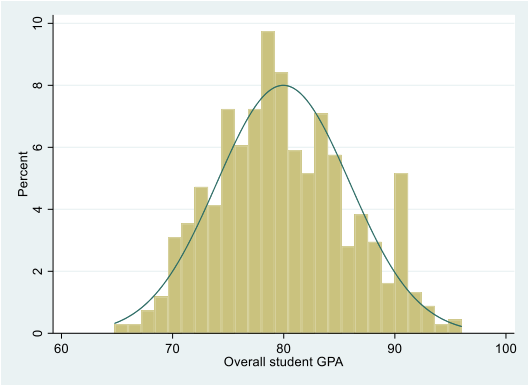


The **fcolor()** option controls the color with which the bars are filled. The **lcolor()** option controls the color with which the bar lines are drawn, and the **lwidth()** option controls the width with which the bar lines are drawn. If you want a list of the colors that you can use, just type:

graph query colorstyle

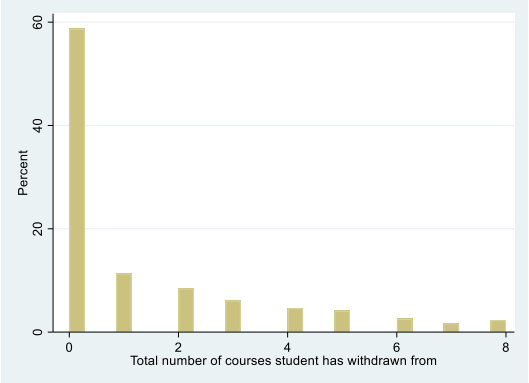
Usually with continuous variables, we would be interested in knowing if the variable resembles a normal distribution. In many cases this is important because some statistical tests assume that the variable is normal. To allow us to compare the distribution of our variable to a normal distribution, Stata allows us to include both on the same plot by using the **normal** option:

histogram gpa, percent normal



As before, the bars represent the distribution of the variable *gpa*, but the line now draws a normal curve on top of the distribution. Looking at the graph we see that the distribution of *gpa* is actually very close to that of a normal distribution. The same, however, cannot be said about another one of our variables, which is the variable *withdraw*:

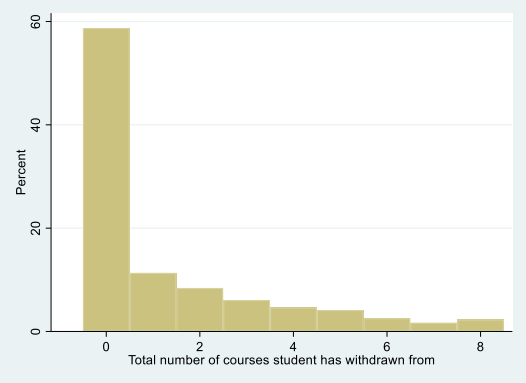
histogram withdraw, percent



As you recall, the variable *withdraw* records the total number of courses that the student has withdraw from so far. The graph shows that most students haven't withdrawn from any courses, followed by a single withdrawal. The distribution clearly does not resemble a normal distribution. There is, however, another difference between this variable and the variable *gpa*. This difference is actually why the histogram of *withdraw* looks ugly. Unlike the variable *gpa*, *withdraw* cannot

take on decimal values. It can only take integer values. Students cannot withdraw from 1.2 courses. They either have withdrawn from one course or from two. This is why the spacing between the bars looks off. Therefore, we need to tell Stata to take into account the fact that this variable cannot take on decimal values. This will affect how the width of the bars are calculated:

histogram withdraw, percent discrete



Notice now that the bar widths make more sense. If a variable cannot take on decimal values, then the width of each bar will be set to 1. This way, each bar will only include one single integer value, with the bars being centered around these integers.