I know that the title of this section is inspecting numeric values, however, the first command that you should use when inspecting a variable whether it was numeric or not is the **codebook** command. This command gives you a very nice summary of the variable. Let's see what it has to say about the numeric variable *gpa*:

codebook gpa

When we run the command, the output tells us the type of the data, which is a double in this case, and the unique values that the variable takes in our dataset. In our case, we see that *gpa* takes on 182 different values. The reason this is possible is that in our dataset the variable *gpa* has entries with two decimal points. So a *gpa* of 75.03 is different from a *gpa* of 75.45. The output actually tells us that the units for this variable is 0.01. Two very important pieces of information about *gpa* is that the mean is 79.9501 and the standard deviation is 5.97769. What does this mean? Well, the mean is basically the average of all observations. So now we know that the average student in the dataset has a *gpa* of 79.9501. The standard deviation is very important because it tells us about the variation that exists in the values of the variable *gpa* relative to its average. For example, if two students take an exam and one scores a 73 and the other scores a 77, the average will be 75. If however the first scores a 60 and the second scores a 90, then the average will also be 75. But both situations are clearly not the same. The variation in the grades in comparison to the average in the second case is much higher than in the first case. This is why reporting the average is not enough. While the average tells us about the central tendency of our data, the standard deviation tells us about the variation of the data. Another example might help illustrate this point. If I told you that the average depth of a certain river is 1.3m and that the standard deviation is 0.1m, then even if you don't know how to swim, you would probably be safe in trying to walk through it, assuming that your height is 1.7m. But if I told you that the average depth of the swimming pool was 1.3m while the standard deviation was 0.9m, then walking through it would not be a good idea since you know that there are certain parts of the pool where the depth is very different from the average.

Going back to our variable, we see that the average is around 79 and that the standard deviation is almost 6. This means that the grades have a strong tendency to be located more or less near the average. Can we visualize this tendency? Sure. We can ask Stata to produce a histogram of the variable. Histograms are one of the best ways to visually see what sort of values a variable takes and how frequently does it take these values. Run the following command:

histogram gpa

Stata produces a nice graph that shows us that most entries have a *gpa* that is between 70 and 90. The graph also allows us to see that the most frequent values are the ones which are close to the mean.

Going back to the output of the **codebook** command, Stata also tells us that there are 142 missing values in the 818 records. In the previous section we saw how to search for observations with missing values. The **codebook** command tells us the exact number of missing values.

We can also see that the command tells us the range of values that the variable *gpa* takes. The smalles value of goa in our dataset is 64.79 and the largest value is 96. You want to test this? Tell Stata to list all observations with a *gpa* that is less than 64.79:

list if gpa < 64.79

No observations are listed. This is because the minimum value is 64.79 which means that no observation has a smaller value.

Finally, the **codebook** command gives us information about something that is called percentiles. Percentiles basically tell us what percent of the observations have a *gpa* that is smaller than a certain value. From the output we see that 10% of observations have a *gpa* that is less than 72, that 25% of the observations have a *gpa* that is less than 75.45, and that 50% of the observations have a *gpa* that is less than 80.

As you can see, the command **codebook** does a very good job summarizing numeric variables. There is another command that gives us similar information and more, and it is the **summarize** command:

summarize gpa

If you run this command, Stata produces a nice table containing several numbers. The first number is the number of observations. This is strange. It says here that there are 676. This is because there are 676 observations where the variable *gpa* is not missing. To make sure, if you subtract 818 and 142 you get 676. When Stata calculates summary statistics of a variable, it ignores missing values of the variable. This is why Stata is telling us that there are 676 observations. It has used only 676 observations to calculate the summary statistics. We see that the table contains the mean and the standard deviation. We also see that these values are more precise than the ones produced by the **codebook** command since they contain more digits after the decimal point. We also get the minimum and the maximum values.

So why use this command? All of this information is available in the **codebook** command. There are three reasons. The first reason is that the command **summarize** has a nice option which is called **detail**. Try the following:

summarize gpa, detail

Stata now gives us more detailed information about the variable. The output gives us more percentiles than those included in the output of the command **codebook**. Another nice set of information is the largest and smallest values. We can see in the output that the smallest values of *gpa* in the dataset are 64.79, 65.8, 66.47, and 66.67. We also see that the largest values of *gpa* are 96, 95.44, 95.44, 94.25.

We can also see in the output that Stata displays three other statistics which are the variance, skewness, and Kurtosis. The variance is basically the square of the standard deviation. This means that it is also a measure of dispersion. The larger it is, the more dispersed the data around the mean.

The smaller its value, the closer the data is to the mean. To confirm that the variance is the square of the standard deviation, we can use Stata's built in calculator:

display 5.977691^2

The display command tells Stata to display something. In this case, we are telling Stata to display the square of the standard deviation. The value that is displayed is exactly that of the variance.

Although this course is not about statistics, I think that it is instructive if you understood what skewness is. Skewness tells us about the symmetry of the data. A skewness of zero means that 50% of the observations are less than the mean and that 50% are more than the mean. A positive value of skewness means that there are more observations with a *gpa* value that is greater than the mean, which as you recall is 79.95. A negative skewness means that there are more observations with a *gpa* value that is less than the mean. In our dataset, the value for skewness is around 0.18, which is very close to zero. This means that the observations are equally distributed to the right and to the left of the mean.

As you see, the **summarize** command with the **detail** option provides us with important information about our variable. There are however as I have already mentioned two other reasons why this command is very useful. The second reason is that this command can be used with the **by** prefix. One of the primary uses of statistics and even data analytics is to compare different groups of individuals. For example, in our dataset, we now know that the average is 79.95. What if I wanted to see the average for different groups of students? What if I was interested in knowing the average of females and of males separately? This can be easily done in Stata. All we have to do is to tell Stata to produce the results per group, or using Stata's language **by** each group. Run the following command:

by gender: summarize gpa

You will notice that Stata gives you an error informing you that the data is not sorted. This is because in order to use the **by** prefix, the data must be sorted by the variable which we will use to divide the data into groups, which is gender is our case. We modify our command as such:

by gender, sort: summarize gpa

Stata now produces the same table it had previously produced, only three times. The first table includes the statistics when only the male students are taken into account, the second table includes the statistics when only female students are taken into account, and the third table produces the statistics for those observations which have a missing value of the variable *gender*. We can see from the output that there are 399 male and 266 females. We can also see that females have a higher average *gpa* than males. We also see that a male student has the lowest *gpa* and that a females student has the highest *gpa*. It is also possible to visualize these results thanks to the fact that the **histogram** command allows us to use the **by** as an option:

histogram gpa, by(gender)

We can see from the output that the value of *gpa* values for females tend to be to the right of 80 while the *gpa* value for males tend to be slightly to the left.

The final reason why the **summarize** command is very useful is that when we use it to list the summary statistics of several variables, we get a nice and clean table. Try the following:

summarize gpa ethical

I don't want to go into the detail of the output here since the variable *ethical* has a different nature than the variable *gpa*. The only purpose of this exercise is for the user to see that the output produced is very easy to read when we are inspecting the summary statistics of more than one variable at the same time. The next section is dedicated to looking at categorical variables like the variable *ethical*.