An Introduction to Stata – Section 4.3
Najib Mozahem

In the last section, we saw what the *_n* variable was and how it was used to list certain observations. It is important to note that all of the commands that we have used did not even contain the variable *_n*. This was done on purpose since the goal of the previous section was for the student to understand that each observation is assigned a number and that these numbers are stored in the *_n* variable. We will use the *_n* variable explicitly in this section. To do that, we will have to learn about another qualifier, which is *if*. What if I wanted to list the observation with a *_n* value of 233? This can be accomplished by running the command:

list in 233/233

This command tells Stata to list the observations starting with the 233rd one and ending with the 233rd one, which practically just includes a single observation. However, a much more intuitive way would be to use the *if* qualifier as such:

list if _n == 233

This command tells Stata to list the observation if the _n value of the observation was equal to 233. Notice that the equal sign is written twice. This is the syntax that you need to use when you use the equal sign with the *if* qualifier. We can also tell Stata to list all observations with a *_n* value that is less than 10:

list if _n < 10

The first nine observations fulfill the requirement because their *_n* values are less than 10. Notice that the tenth observation is not displayed, since it has a *_n* value of 10, and 10 is not less than 10. The following command will display the 10th observation together with the first nine:

list if _n <= 10

This command is telling Stata to list all observations with a *_n* value that is less than or equal to 10. Since 10 is equal to 10, the tenth observation fulfils the criteria and hence is listed.

The *if* qualifier can be used without the *_n* variable. For example, what if we wanted to list the observations of those students who have a GPA that is less than 68? We can do this by executing the following command:

list if gpa < 68

We see that there are only four observations that are listed. What if we wanted to list students with a GPA that is exactly equal to 69? We can do that by executing the following:

list if gpa == 69

Notice that only two students have a *gpa* that is exactly 69. Also remember that we use two equal signs when testing for equality with the *if* qualifier. We can also use the *if* qualifier to look at observations with missing values. If you recall, missing values are stored as a "." In Stata. Therefore, the following observation will tell Stata to list all observations where the *gpa* value is missing:

```
list if gpa == .
```

Notice that the *gpa* of all the listed observations is missing. We could have performed the above operation by taking advantage of a nice function:

```
list if missing(gpa)
```

This command tells Stata to list the observations where the *gpa* variable is missing. Someone might argue that it would make more sense to list the observations without a missing value of the variable. This can also be done in Stata like this:

```
list if !missing(gpa)
```

The "!" character is read as "not", so the command is telling Stata to list the observations if the gpa is not missing. The "!" character can also be used without the missing function. For example, you can tell Stata to list the observations where the *gpa* is not equal to 69:

```
list if gpa != 69
```

Notice here that we only write one equal sign.

Next, I want you to try to run the following command:

```
list if gpa >= 96
```

We are telling Stata to list the observations where the *gpa* is greater than or equal to 96. However, we see in the output that Stata has also listed observations where the *gpa* is missing. This is because a missing value in Stata is considered to be larger than any number. GPAs should not exceed 100 right? Try to execute the following:

```
list if gpa > 200
```

You will see that Stata displays many records and that all of these records have a missing value for the *gpa* variable. If we wanted Stata to list the observations with a *gpa* that is greater than or equal to 96 but that are not missing, we can use the following command:

```
list if gpa >= 96 & gpa < .
```

We see that Stata lists a single observation. Observations with missing *gpa* values were not listed. The reason is that the command uses the "&" which Stata considers to be an AND operator. So the command is telling Stata to list all observations who have a *gpa* that is greater than or equal to 96 AND who have a *gpa* which is less than "." Stata will only list the observations that satisfy both requirements.

Another operator that is commonly used is the OR operator, which is represented by the character "|". While the AND operator tells Stata to list the observations that satisfy both conditions, the OR operator tells Stata to list the observations that satisfy either conditions. For example, try running the following command:

```
list if gpa <= 66 | gpa > 95
```

You will see that Stata displays all observations that either have a *gpa* that is less or equal to 66 or that have a *gpa* that is greater than 95. So we are basically telling Stata to list students with a very high *gpa* and those with a very low *gpa*. Unfortunately, you can also see that Stata displayed students with a missing value of *gpa*, and this is clearly not what we wanted. This means that we need to tell Stata to list students with a *gpa* value that is not missing. This is performed by the following command:

list if (gpa <= 66 | gpa > 95) & !missing(gpa)

Although the command looks complicated, it is in fact quite simple. The most important thing to note in the command is the use of the parentheses. Logically speaking, we want to list observations that satisfy two conditions. The first condition is that the *gpa* is less than or equal to 66 OR the *gpa* is greater than 95. The second condition is that the *gpa* should not be missing. The "&" operator is placed between both conditions that must be satisfied. The first condition is made up of multiple parts, so we need to put it in parentheses in order to tell Stata that these parts belong with each other. This is just like simple algebra. If you have (A + B) X C, then first you add A and B and then you multiply C. In the above command, Stata gets all observations that have a *gpa* that is less than or equal to 66 OR a *gpa* that is greater than 95. Stata then only lists the observations in the last that do not have a missing value for the variable *gpa*.

In this section we have only used the operator with the variable *gpa*, and not with other variables. The reason is that we will see how the operator works with other types of variables after we cover categorical variables in more detail in section 5.