Often, you will need to load an external data file into R to do some analysis and modeling.

In this class, we'll be working with csv files, or comma separated value files.

This is a common format for data files and is easy to work with in R.

The first thing you need to do to read in a data file is to navigate to the directory on your computer where the data file is saved.

On a Mac, go to the Misc menu, then select "Change Working Directory...".

On a PC, go to the File menu and select "Change dir...".

This should pop up a browsing or navigation window.

Navigate to the folder where you saved the data file WHO.csv that you've downloaded for this class, and then select that folder.

Nothing should have happened in R, but if you type getwd, and then empty parentheses and hit Enter, you should see the path to the folder containing the data set that you just selected.

Now, read in the data file by typing WHO = read.csv("WHO.csv") the name of the data file we want to read in.

If you hit Enter, this will save the data set in WHO.csv to the data frame WHO.

To look at our data, there are two very useful commands.

The first is the str function, which shows us the structure of the data.

If you type str(WHO), and hit Enter, you can see that we have a data frame of 194 observations and 13 variables.

This data set contains recent statistics from the World Health Organization-- W, H, O, or WHO-- on all countries.

The variables are the name of the country, the region the country is in, the population in thousands, the percentage of the population under 15 and over 60, the fertility rate or average number of children per woman, the life expectancy in years, the child mortality rate which is the number of children who die by age five per 1,000 births, the number of cellular subscribers per 100 population, the literacy rate among adults aged greater than or equal to 15, the gross national income per capita, the percentage of male children enrolled in primary school, and the percentage of female children enrolled in primary school.

For each variable, str gives us the name of the variable, and then a description of the type of the variable followed

by a first few values of the variable.

We see a couple different types here.

One is a factor variable.

Country and Region are both factor variables.

This means that the variables have several different categories, not necessarily numerical.

For example, the Region variable has six different categories or levels.

These include Africa and Americas.

So each observation in the Region variable belongs to one of six different categories.

For variables like Country, where there's 194 levels, which is the same number of observations we have, each value in this variable is different.

In this case, it makes sense, since each country name is different.

Then we have two types of numerical values-- integer and then general numerical values.

The other very useful function to take a look at our data is the summary function.

In your R console, type summary and then, in parentheses, WHO, the name of our data frame, and hit Enter.

This gives a numerical summary of each of our variables.

For the factor variables, country and region, it counts the number of observations in each of the levels or categories.

So here, we see that we have 46 countries in the region Africa, 35 in the region Americas, etc.

For each of the numerical values, we see the min, first quartile, median, mean, third quartile, and maximum values in that variable.

We can also see in some of the variables that we have this category called NA's.

This means that some observations are missing values in that variable.

So for FertilityRate, there 11 observations that are missing the value of FertilityRate.

When working with data in R, it can often be useful to subset your data.

For example, suppose we want to create a new data frame with only the countries in Europe.

Let's call it WHO_Europe and use the subset function to subset the data frame WHO to take only the observations for which Region is exactly equal to Europe.

The subset function takes two arguments.

The first is the data frame we want to take a subset of, in this case, WHO.

And the second argument is the criteria for which observations of WHO should belong in our new data frame, WHO_Europe.

In this case, we want the observations for which the Region variable is exactly equal to Europe.

The double equal sign here means exactly equal to.

If we hit Enter and then look at the structure of WHO_Europe, we can see that we now have a data frame of 53 observations of the same 13 variables.

Does 53 sound right?

Well, let's look back at the summary output of WHO.

We can see in the Region output, there were 53 observations that belonged in the region Europe.

So we should expect 53 observations in our Europe subset, which is right.

Now, suppose we want to save this new data frame, WHO_Europe, to a csv file.

You can use the write.csv function to do this.

Type write.csv, and then in parentheses the name of the data frame we want to save, WHO_Europe, comma, and then in quotes the name of the file we want to save it to.

Let's call it WHO_Europe.csv.

If you hit Enter, nothing should happen, but you should now have a file called WHO_Europe.csv in the same folder that you saved WHO.csv in.

And now that we've saved this as a csv file, if we're not working with it anymore in R, we can remove the data frame from our current session in R.

This is often useful if you're working with a large data set that's taking up a lot of space.

First, let's type ls() to see what variables we currently have in R. You could see that WHO_Europe is one of our variables.

Now, type rm for remove and then the name WHO_Europe and hit Enter.

If you type ls() again, you should see that WHO_Europe is gone.

In the next video, we'll explore the WHO data set.