

MITOCW | MIT15_071S17_Session_1.3.06_300k

So far, all of our output values and variables have been single numbers.

You can also create more advanced data structures in R like vectors and data frames.

A vector is a series of numbers or characters stored as the same object.

You can create a vector in R using the `c` function, which stands for combine.

In your R console, type `c(2,3,5,8,13)` and hit Enter.

This creates a vector of five numbers all stored as the same object.

To learn more about vectors in R, let's enter some data into R about the life expectancies in different countries.

We'll first create a vector of country names called `Country` using the `c` function.

We'll put each country name in quotes since we are typing characters not numbers.

So in your R console, type `Country` equals and then `c` for the combine function, parentheses and then `"Brazil"`, `"China"`, `"India"`, `"Switzerland"`, and then `"USA"`.

Close the parentheses and hit Enter.

Now, let's create a second vector.

This time with the life expectancies of these five countries in the same order that we entered the country names.

We'll call this one `LifeExpectancy = c(74,76,65,83,79)` and hit Enter.

Now if you take a look at both `Country` and `LifeExpectancy`, you can see that `Country` has five character values and `LifeExpectancy` has five numerical values.

A word of warning-- you shouldn't try to combine characters and numbers in the same vector.

R only allows one data type in each vector so all of the numbers will be converted to characters.

This is bad because it won't allow us to do any numerical calculations with the numbers, like compute the mean.

If you want to display an element of a vector, use square brackets.

For example, we could type `Country[1]` to get the first element of the `Country` vector, `Brazil`.

Or we could type `LifeExpectancyLifeExpectancy[3]` to get the third element of `LifeExpectancy`, 65, corresponding to India.

Another nice function to create vectors in R is the `seq` function which creates a sequence of numbers.

Try typing `seq(0,100,2)`.

Close the parentheses and hit Enter.

This gives a sequence of numbers from 0 to 100, starting at zero, the first argument to the `seq` function, ending at 100, the second argument to the `seq` function, and the numbers are in increments of two, the third argument to the `seq` function.

This can be useful if you want to create a unique identifier for observations.

Now, let's combine our vectors into what we call a data frame in R. This will be an important data structure for us because all of the data files we'll work with in this course will be loaded as data frames.

Additionally, many algorithms in R require all of the data to be in a single object like a data frame.

We'll call our data frame `CountryData` and then use the `data.frame` function to combine `Country` and `LifeExpectancy`.

So after typing `CountryData = data.frame`, in parentheses type `Country` comma `LifeExpectancy`.

Then close the parentheses and hit Enter.

Let's take a look at `CountryData` by typing `CountryData` and hitting Enter.

It has two columns, our variables, and five rows, our observations.

It's similar to how you might organize data in a spreadsheet software like Excel.

Let's say we now want to add another variable to our data frame-- the population in thousands of each country.

We can do this by using a dollar sign to link the new data into the data frame.

So we'll type the name of our data frame, `CountryData`, and then a dollar sign followed by the name of the new variable we want to add, `Population`.

Then we'll set this equal to `c`, and in parentheses the population in thousands of each country.

So 199000, 1390000, 1240000, 7997, and 318000.

Close the parentheses and hit Enter.

Now if you take a look at CountryData you should see that we have a third column, Population.

We'll use this dollar sign notation a lot and we'll talk about it more later in this lecture.

Note that we had to give the population values in the correct order.

R will just combine the vectors in the order they're typed.

Now, suppose we want to add two new observations for Australia and Greece.

We first need to create new Country, LifeExpectancy and Population vectors.

So we will now set Country equal to the names of the new countries, Australia and Greece.

We'll set LifeExpectancy equal to the new life expectancies, 82 and 81, and we'll call Population a vector of the populations of these countries 23050 and 11125.

Then we can create a new data frame.

We'll call this one NewCountryData and we'll set it equal to `data.frame(Country, LifeExpectancy, Population)`.

Note that we combined three vectors here with the `data.frame` function.

If we take a look at NewCountryData, we can see that it's very similar to CountryData.

Lastly, let's combine CountryData and NewCountryData with the `rbind` function which combines data frames by stacking the rows.

We'll call this new data frame AllCountryData and we'll use the `rbind` function to combine CountryData, and NewCountryData.

If we take a look at AllCountryData we can see that it has the three variables from before and all seven observations.

The functions we've used in this video have allowed us to create data structures and modify data structures in R. But most of the time you'll be reading in data from an external file, which we'll do in the next video.