We’re going to start with recapping some base R and dplyr functions that you may have already come across. Dplyr is part of the tidyverse, so I’m loading that, and also the NHSRdatasets package, which includes the data we’re going to use.

To run multiple lines of code in the source pane we can select the rows we want & press Ctrl + Enter.

Next we load some data for attendances at Accident & Emergency (A&E )

To run a single line of code we can position the cursor anywhere in the line and use Ctrl + Enter.

In the environment panel in the top right corner, by clicking the blue arrow we can see the structure of the dataframe – over 12,000 rows of 6 columns, and it tells us the data type of those columns.

We can also click on the symbol on the right which opens the dataframe in a tab in the source pane.

To get the names of the columns in a dataframe we can use either colnames or names function.

* colnames(data) # works only with matrices and dataframes
* names(data) # works only with vectors and dataframes

Since we’re working with a dataframe, they both give the same output.

Using dataframe name $ column name will return the content of that column, but especially when it’s a long column, it can be more useful to just find the unique data items. Now we can see there are 274 unique entries in that column.

If you know the position of a specific entry you want to retrieve, you can use square brackets. For a range of entries using the colon. \*If you’re coming to R with a knowledge of python, note that R starts counting from 1, not zero.

Range is a really useful function for working with numeric or character data, returning the maximum and minimum values.

STR function shows you the structure of the dataframe, the same as you can see in the environment, but returns it as an output down in the console.

The head & tail functions allow you to see a snippet of the dataframe, by default head() returns the first 5 rows and tail() the last 5 rows. By adding a number after the data source we can change that and return however many rows we’d like to see.

Dplyr has a group of similar functions called slice\_. Not only can you get a number of rows, you can also choose a proportion. If you want to order the dataframe by a different column, you can add order\_by = .

There’s lots more information on the various slice\_ functions in the help file, which you can access using ?slice in the console pane.