Today you’ll practice performing the t test and F test and interpreting the results. You will also learn another type of plot – the box-and-whisker plot – that is suitable for graphically summarizing continuous data that is divided into groups.

Part 1: the t test

Remember that the purpose of the t test is to test for differences in the means of two samples (and only two samples). To run the test, you will need to provide two numeric vector variables as the inputs. These vectors will almost certainly be the result of a subset command. R performs a version of the t test called Welch’s t test. This version is superior to the more traditional Student’s t test, but you should report the specific type of test when you write your results.

The command (for a two-tailed Welch’s t test) is simple:

```
t.test(variable1, variable2)
```

If you want to perform a one-tailed test (Hₐ is “greater than” or “less than”):

```
t.test(variable1, variable2, alternative = "g")
```

Alternative=“g” (or “greater”) is for Hₐ that variable 1 is greater than variable 2
Alternative=“l” (or “less”) is for Hₐ that variable 1 is less than variable 2

Part 2: the F test

The F test also requires two and only two vector variables as an input, so the command is similarly straightforward.

```
var.test(variable1, variable2)
```

Like with the t test, you can also perform one-tailed F tests with alternative=“g” or “l” (but you are unlikely to do so, unless you have strong prior evidence for that alternative hypothesis).

Part 3: Box-and-whisker plots

If your data and question leads you to the t or F test, a box-and-whisker plot is a good way of summarizing that data graphically. In the default box-and-whisker plot, the thick line is the median value. The box spans the interquartile range, which is the range from the 25th percentile to 75th percentile of the data. In a data set with 100 values, the 25th percentile is the 25th smallest and the 75th percentile is the 75th smallest (or 25th largest). The whiskers extend 1.5 times the interquartile range above the top and below the bottom of the box. Any data outside of that distance will be shown as points. An example is shown on the next page.

R is somewhat intelligent, so if you use the generic plot function and provide a factor variable as the x coordinate (the first argument) and a numeric variable as the y coordinate (the second argument), it will automatically plot a box-and-whisker plot using the levels of the factor variable as the categories.
For example:

```
plot(georoc$rock.type, georoc$SiO2)
```

R knows that the generic plot function refers specifically to the function `boxplot()` in this case. To see the help for box-and-whisker plots (including many additional options), you can type `?boxplot` at the prompt.

When you make a plot, R will assign default axis labels (in the case of boxplots, they will be blank). When using the plot function, you must specify the x and y coordinates first, but after that you can include a variety of options, in any order as long as they are separated by commas. Two of those options are:

```
xlab = "..."  
ylab = "..."
```

They are used to specify the x and y axis labels.

There is one complication with generating box-and-whisker plots. The plot bases the categories on the levels of the factor variable. You have learned how to subset a data frame to select only certain values (for example, only rhyolite rock type). However, even though the new data frame only contains rhyolites in the rock.type column, all four original levels are still retained (three are just empty). A box-and-whisker plot will include blank spaces for those empty levels. To make the box-and-whisker plot work (and other functions that use factor levels), you must reset the levels by re-making the column into a new factor.

```
new_data_frame$column <- factor(new_data_frame$column)
```

Alternatively, if you want to reset the levels in all factor columns within a data frame, you can use:

```
new_data_frame <- droplevels(new_data_frame)
```