Today you’ll practice performing ANOVA and interpreting the results. You will also have more time to practice working with subsets, factor levels, and making box-and-whisker plots.

**ANOVA**

Because ANOVA can have an unspecified number of input variables (>2), the syntax is different. You need one column with continuous numerical data (a vector variable) – the measurements – and one column that contains names of categories (a factor variable) that indicate the samples.

The function is called aov. Note that you should store the results of the ANOVA as a variable to actually see meaningful information:

You can either write (the second is preferred):

```
aov(dataframe$measurements ~ dataframe$categories)
```

```
aov(measurements ~ categories, data = variable)
```

The “∼” symbol (a tilde) means “as a function of” in R.

To see the results, type:

```
summary(aov_results)  # substitute the variable name you chose
```

**Tukey Test**

In R, the post-hoc Tukey test must be performed on the results of a previous ANOVA. This is also why you should store the output from the aov function as a variable.

```
TukeyHSD(aov_results)  # remember case sensitive TukeyHSD!
```

**Optional: advanced data manipulation**

If you are confident about your skills with subset, here are some more advanced functions for data manipulation. You won’t need to use these right now (subset is enough for our purposes at this point), but they can come in handy when trying to find the mean/median/sd in many sample categories.

The function `split()` splits a data frame (or specified columns) into a list variable, using the levels from a factor variable. The list elements are taken from the factor. Try this and look at the list output:

```
split(georoc$SIO2, georoc$tectonic.setting)
```
The function splits the SiO2 data into a list with categories taken from the factor levels (the unique names) in the tectonic setting column.

Save the output of the split function as a variable called sio2_split.

That list format is extremely useful because R contains powerful functions (called the “apply family”) that operate on lists in a parallel way. Instead of creating a loop or running the same command multiple times, there is a single function that will apply one command simultaneously to all elements in the list. Try this:

```
sapply(sio2_split, mean)
```

Note that functions can be nested in R, so you could just use:

```
sapply(split(georoc$SIO2, georoc$tectonic.setting), mean)
```

You can also use the function by(), which has this syntax:

```
by(measurements, categories, function)
```

For example:

```
by(georoc$SIO2, georoc$rock.type, mean)
```