Table 1: How to fix some common errors in R

Error message	How to fix it
`there is no package called 'blah'`	You have either mis-typed the package name or the package is not installed. Check the spelling or use `install.packages()` to install the package.
`could not find function "blah"`	You have either mis-typed the function name or the package containing that function is not loaded. Check the spelling or use `library()` to load the package.
`object 'blah' not found`	You have either mis-typed the name of the object or the object does not exist. Check the spelling and make sure you have run the code that creates the object. This error can also occur when you have forgotten to put quotes around a character value in an argument, since R treats words without quotes around them as the names of objects. In this case, check that you have used quote marks around any character values in your code.
`'blah' does not exist in current working directory`	You have either mis-typed the name of a file that R is trying to access or the file does not exist in the location you have specified. Check the slepping and make sure the file exists.
`non-numeric argument to binary operator`	You have tried to use a mathematical operator such as `+` or `-` with a non-numeric value. For example, you might have written the code `1 + blah` thinking that `blah` holds a numeric value, but if `blah` actually holds a character value then trying to add it to `1` makes no sense. Check that any objects in your code have the values you expect them to.
`object of type 'closure' is not subsettable`	You have tried to use a function as if it is an object, which can happen when you store data in an object that has the same name as an R function. The circumstances that produce this error are often not simple to understand, so the best way to handle this error is to avoid it by not naming objects using the names of functions.
`no applicable method for 'blah' applied to an object of class "blah"`	Some R functions (called generic functions) work in different ways depending on what type of object you use them on. But if you use a generic function on an object that it does not know how to handle, you will see this error. Check the object that you provided to the function causing the error to make sure it is the type of object you are expecting it to be.
`unexpected numeric constant in "blah"`, `unexpected string constant in "blah"` or `unexpected symbol in "blah"`	You have a typo somewhere in your code. Check the line of code producing the error to make sure it is formatted correctly. The most common typos that cause this error are a missing comma or closing parenthesis, but there are several other typos that can cause similar errors.
`unused argument (var = "blah")`	You have used an argument name in a function that does not understand it. Check the manual page for that function.
`argument "blah" is missing, with no default`	You have used a function without providing all the necessary arguments. Check the manual page for that function.