Lab 9, Week 12

Sidi Wu

As in lab 8, you should be working from your `mars` project for this lab. You will need the `testfwd_stepwise.RData` and `testbwd_stepwise.RData` files from the `tests/testthat/` folder of your `mars` project. In addition, copy the file `testmars.RData` from the `Exercises/ProjectTestfiles/testthat` folder of the class GitHub repository to your `tests/testthat` folder.

1. Finish your `mars()` function by (i) fitting the final model to the basis functions returned by `bwd_stepwise`, (ii) packaging your output as a list in the following order (to enable tests against my output):
   - `call`: function call
   - `formula`: formula input by user
   - `y`: response variable
   - `B`: final set of basis functions for the input data, returned by `bwd_stepwise()`
   - `Bfuncs`: The `Bfuncs` returned by `bwd_stepwise()`
   - `x_names`: The colnames of the model matrix constructed from the input formula and data. These will be needed by our `mars.summary()` function to give names to the variables indicated in `Bfuncs`.
   - The rest of the list output by `lm` from part (i). Finally, give your output list class `mars` that inherits from class `lm`.
2. Create test scripts `testbwd_stepwise.R`, and `testmars.R` in your `mars/tests/testthat` folder. Run `devtools::test()` on your package and make sure it passes all tests.
   - `testbwd_stepwise.R` should load `testbwd_stepwise.RData`, call your `bwd_stepwise()` function with inputs `testfwd` and `testmc`, and use `expect_equal()` to compare the output to the output `testbwd` from `testbwd_stepwise.RData`
   - `testmars.R` should load `testmars.RData`, call your `mars()` function with the formula `y~.`, data `marstestdata`, and control `testmc`, and use `expect_equal()` to compare the output to `testmars`. **Note:** Add the argument `ignore_attr=TRUE` to your call to `expect_equal()`. The output of `mars()` includes formulas and other objects that store the environment in which they were created in their attributes. These environments will be different from one call of `mars()` to the next.